AD-A114 068 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/G 17/2
DEVELOPMENT OF A GOOD-QUALITY SPEECH CODER FOR TRANSMISSION OVE--ETC(U)
MAR 82 V R VISWANATHAN, M BEROUTI, A HIGGINS DCA100-80-C-0039
UNCLASSIFIED BBN-4916 NL

1 of 2
ADA
114068

Bolt Beranek and Newman Inc.

12

Report No. 4916

Development of a Good-Quality Speech Coder for Transmission Over Noisy Channels at 2.4 kb/s

Final Report

March 1982

Prepared for:
Defense Communications Agency

12

Report No. 4916

DEVELOPMENT OF A GOOD-QUALITY SPEECH CODER FOR TRANSMISSION OVER NOISY CHANNELS AT 2.4 kb/s

Final Report

Authors: V.R. Viswanathan, M. Berouti, A. Higgins, and W. Russell

March 1982

Prepared for:

Defense Communications Agency

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER BBN Report No. 4916	2. GOVT ACCESSION NO. AD-A114 068	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DEVELOPMENT OF A GOOD-QUALITY SPEECH CODER FOR TRANSMISSION OVER NOISY CHANNELS AT 2.4 kb/s		5. TYPE OF REPORT & PERIOD COVERED Final Report Aug. 1980 - March 1982
		6. PERFORMING ORG. REPORT NUMBER BBN Report No. 4916
7. AUTHOR(s) V.R. Viswanathan, M. Berouti, A. Higgins, and W. Russell		8. CONTRACT OR GRANT NUMBER(s) DCA100-80-C-0039
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency Contract Management Division, Code 680 Washington, DC 20305		12. REPORT DATE March 1982
		13. NUMBER OF PAGES 179
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Speech coding, linear predictive coding, 2.4 kb/s speech transmission, narrowband speech coders, harmonic deviations vocoder, variable frame rate transmission, robust speech transmission over noisy channels.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the development, study, and experimental results of a 2.4 kb/s speech coder called harmonic deviations (HDV) vocoder, which transmits good-quality speech over noisy channels with bit-error rates of up to 1%. The HDV coder is based on the linear predictive coding (LPC) vocoder, and it transmits additional information over and above the data transmitted by the LPC vocoder, in the form of deviations

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

between the speech spectrum and the LPC all-pole model spectrum at a selected set of frequencies. At the receiver, the spectral deviations are used to generate the excitation signal for the all-pole synthesis filter.

The report describes and compares several methods for extracting the spectral deviations from the speech signal and for encoding them. To limit the bit-rate of the HDV coder to 2.4 kb/s the report discusses several methods including orthogonal transformation and minimum-mean-square-error scalar quantization of log area ratios, two-stage vector-scalar quantization, and variable frame rate transmission. The report also presents the results of speech-quality optimization of the HDV coder at 2.4 kb/s. The final optimized 2.4 kb/s coder yields noticeable improvement in speech quality over the 2.4 kb/s LPC coder, and it produces only a slight degradation in speech quality and intelligibility, as the channel bit-error rate is increased from 0% to 1%.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

The authors wish to acknowledge the help of the following individuals: B. Atal, K. Field, K. Karnofsky, M. Krasner, J. Makhoul, G. Moran, S. Roucos, and R. Schwartz.

LIST OF FIGURES

LIST OF FIGURES (CONT'D)

LIST OF TABLES

1. INTRODUCTION

The overall objective of this project was the development and optimization of a speech coding algorithm that produces good-quality speech at a data rate of 2.4 kb/s (kilobits/second). In this chapter, we state the specific requirements on the coder performance (Section 1.1), describe briefly the optimized coder (Section 1.2), and provide an overview of the rest of this report (Section 1.3).

1.1 Coder Performance Requirements

Performance requirements on the narrowband speech coder include the following:

1. Synchronous data transmission: Transmit data at a synchronous (or fixed) rate of 2.4 kb/s.

2. Improvement over LPC-1Ø: Produce a significant improvement in speech quality over LPC-1Ø, which is the current Department of Defense (DoD) standard for speech transmission at 2.4 kb/s.

3. Noisy channel: Provide highly intelligible speech under the condition of transmission bit error rates of up to 1%.

4. Acoustic background noise: Produce good-quality speech in the presence of acoustic background noise typical in an office environment (sound pressure level of the noise being about 6Ø dB re 2Ø micronewtons per square meter).

5. Tandem operation with CVSD: Perform satisfactorily in

tandem with a 16 kb/s CVSD speech coder. The tandem link must provide negligible loss of intelligibility as compared to the new coder operating in a single link.

1.2 Summary of the Optimized Speech Coder

The speech coding algorithm we chose is based on the linear prediction method and is called the harmonic deviations (HDV) coder. The optimized HDV coder may be summarized as follows. In the transmitter, the analog speech is lowpass filtered at 5 kHz, sampled at 10 kHz, and divided first into frames of 20 ms duration and then into 9-frame blocks. A variable frame rate (VFR) algorithm is used to select and transmit only 6 frames of data every block, along with a block header to identify the transmitted frames. For every frame selected by the VFR algorithm, the following quantities are transmitted: a synchronization bit, voicing status, pitch, speech signal energy, 12 linear predictor coefficients (log area ratios), and 3 selected spectral deviations between the log spectrum of the speech signal in the frame and the log spectrum of the all-pole model. These quantities are quantized, coded, partially error-protected, and transmitted across the channel.

At the receiver, the data for the untransmitted frames are regenerated by linear interpolation between adjacent transmitted frames. The output speech of the coder is synthesized, pitch-synchronously for voiced frames and every 1Ø ms for unvoiced frames, by generating the excitation signal using the spectral deviations and applying it to the all-pole synthesis filter. The digital output is D/A converted and lowpass filtered at 5 kHz to produce the analog output speech.

A non-real-time FORTRAN simulation of the optimized HDV coder was developed, delivered, and demonstrated on the sponsor's PDP-11/34 minicomputer.

The optimized, 2.4 kb/s HDV coder produces a significant speech-quality improvement over the LPC-1Ø coder. The improvement is in the form of reduced "buzziness", "muffled" quality, and background noises and a more natural voice quality. The extent of speech-quality improvement is more for male speakers than for female speakers. When operating over channels that cause 1% random bit-errors, the HDV coder produces some audible but mostly low-level distortions such as "pops" and "clicks" in the output speech; there is only a small difference in speech quality and intelligibility between 1% channel error and noise-free cases.

1.3 Overview of the Report

In Chapter 2, we discuss the methodology that leads to the formulation of the HDV coder. Chapter 3 provides a block-diagram description of the HDV coder and describes the input-speech databases we used during this work. In Chapter 4, we discuss three groups of methods for extracting the spectral deviations and compare them using experimental results. In Chapter 5, we describe our work on pitch-synchronous HDV coders. This work was performed in an effort to study the issues underlying the speech-quality of HDV coders and to develop improved methods for extracting the spectral deviations. Chapter 6 deals with several aspects of synthesis that we investigated with the goal of improving the speech quality of the HDV coder. In the next three chapters, we consider ways of lowering the bit-rate of the HDV coder with only a negligible effect on the quality of the output speech: coding of the spectral deviations (Chapter 7), quantization of the log area ratios (Chapter 8), and variable frame rate transmission of HDV coder data (Chapter 9). The results of our work on speech-quality optimization of HDV coders at 2.4 kb/s are reported in Chapter 10 for error-free channels and in Chapter 11 for noisy channels. In Chapter 11, we also recommend an HDV coder design as being the most robust and best overall 2.4 kb/s coder. The performance of this optimized coder in office background noise is treated in Chapter 12, and its performance in tandem with a CVSD coder is discussed in Chapter 13. Chapter 14 summarizes the details of the final optimized 2.4

kb/s HDV coder. Finally, in Chapter 15, we describe our work on the FORTRAN simulation of the optimized HDV coder on the sponsor's PDP-11/34 minicomputer.

2. FORMULATION OF THE HARMONIC DEVIATIONS VOCODER

Over the last decade or so, significant advances have been made in understanding the properties and workings of the linear prediction method, as well as in optimizing the speech quality of the narrowband LPC vocoder. However, a comparison of the LPC speech with the input natural speech readily reveals the substantial loss of naturalness and lack of "fullness" in LPC speech. Also, LPC speech is particularly degraded for certain speech material and for some speakers (e.g., high-pitched, breathy voices).

These speech quality degradations are not specific to LPC vocoders; they are also perceived at the output of other narrowband vocoders such as channel, homomorphic, and formant vocoders. For our discussions given below, we limit our attention to LPC vocoders, although the general conclusions we make will also apply to other types of vocoders.

With the goal of improving significantly upon the speech quality of LPC vocoders, we first probe into several possible causes of the unnatural speech quality of LPC vocoders in Section 2.1. Following that, we present in Section 2.2 a speech modeling methodology that has served as the basis for the work reported in later chapters.

2.1 Causes of Speech Quality Degradations

As possible causes of speech quality degradations in LPC vocoders, below we inquire into three broad items: quantization accuracy, source model accuracy, and spectral model accuracy.

2.1.1 Quantization Accuracy

In an LPC vocoder, there are three types of quantization:

- Parameter quantization (i.e., quantization of pitch, speech signal energy, and log area ratios)
- Order quantization (i.e., limiting the number of log area ratios transmitted)
- Time quantization (i.e., limiting the frame rate of parameter transmissions).

From previous experience, we know that for each of the three quantization types, as quantization accuracy is progressively increased, perceived speech quality increases significantly at first but levels off subsequently [1, 2]. Clearly, a proper tradeoff of bits among the three quantization processes will maximize the speech quality at a fixed data rate of 2.4 kb/s. But, even an 8.7 kb/s LPC coder with high quantization accuracies (reported in [1]) exhibits most of the unnatural quality that we referred to above. Thus, we conclude that the three quantization processes are not the primary source of the unnatural quality of LPC coders.

2.1.2 Source Model Accuracy

The linear prediction residual signal is the ideal source for the LPC synthesizer in the sense that if it is applied to the all-pole synthesis filter, the output speech becomes identical to the original input speech. However, the residual signal cannot be satisfactorily transmitted over a narrowband channel. Several methods have been suggested in the literature for modeling the excitation signal for use in narrowband vocoders. All these methods, however, lead to varying degrees of speech quality degradations. The commonly used pulse/noise binary source model produces "buzziness" and lack of "fullness" in the synthesized speech, even if there are no errors in the computed pitch or voicing decision. Of course, errors in pitch or voicing cause additional degradations in speech quality and intelligibility. The mixed-source model, which allows for simultaneous pulse and noise excitations, largely eliminates the "buzziness" and reduces the lack of "fullness" [3]. But, the synthesized speech still sounds unnatural when compared to the original speech.

Almost all source models (including the two mentioned above) assume a flat amplitude spectral envelope for the excitation signal. The true short-term spectral envelope of the residual signal, however, often has a dynamic range or spectral spread of about 5 to 10 dB. As an example, Fig. 1 shows the short-term

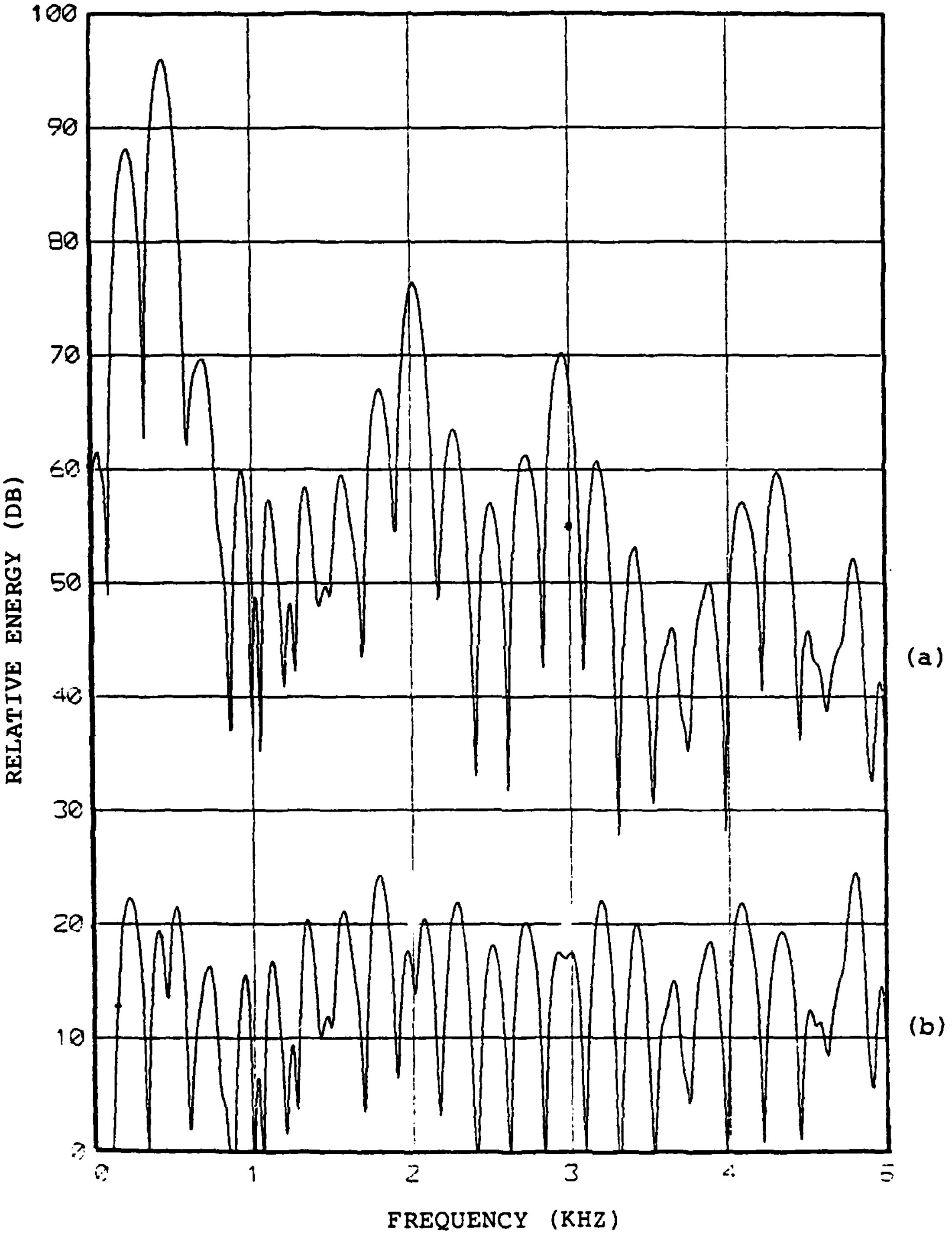


Fig. 1. Illustration of the frequency variations of the spectral envelope of the LPC residual signal.

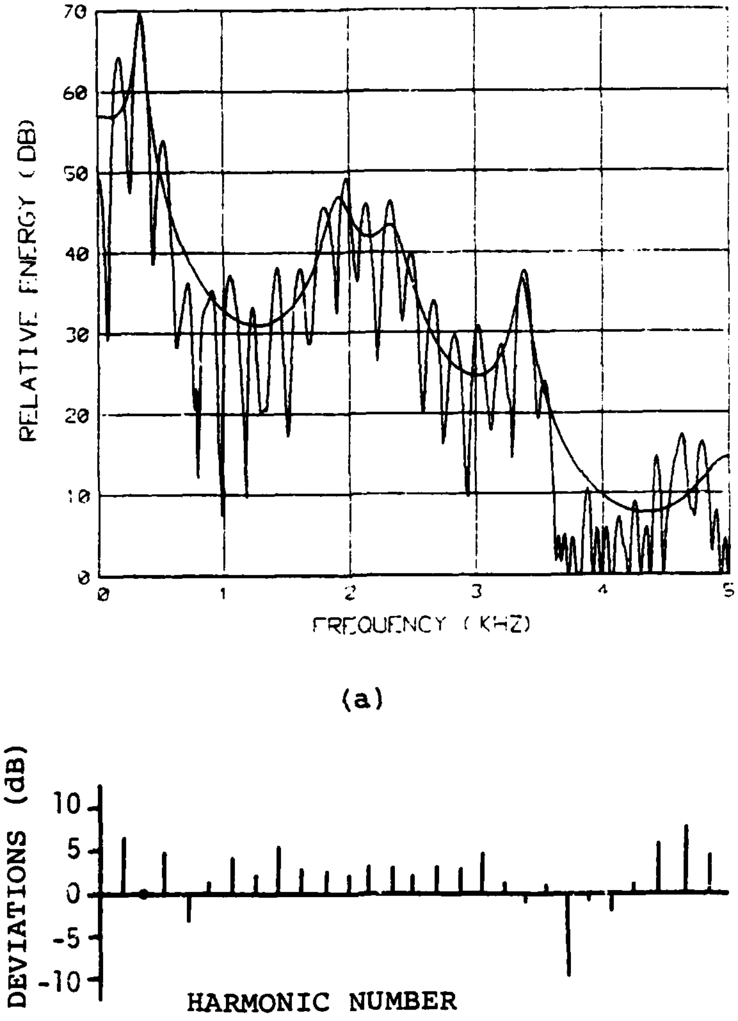
(a) Short-term spectrum of the speech signal.
(b) Short-term spectrum of the residual signal obtained using 12-pole LPC analysis.

spectrum of the residual signal along with the spectrum of the speech signal, for a vowel sound. We believe that the deviations of the spectral amplitudes of the residual signal from a constant value (representing its average energy) are important for natural-quality speech synthesis.

Since the LPC all-pole filter has the minimum phase property, the short-term phase of the synthesized speech is determined by the short-term phase of the excitation signal. While the long-term phase of the residual signal, which affects pitch, must be preserved in the excitation signal, it is generally believed by speech researchers that the short-term phase (within a pitch period) at best plays a secondary role in determining the naturalness of synthetic speech. However, Atal and David have recently reported that speech quality does improve with the introduction of a simple phase model [4].

2.1.3 Spectral Model Accuracy

The spectrum of the LPC all-pole model provides a good approximation to the envelope of the speech signal spectrum [5]. However, this does not ensure that the two spectra have a good agreement at <u>all</u> frequencies. In practice, we observe large differences between the two spectra at several frequencies. This point is illustrated in Figs. 2 and 3. Both examples are the



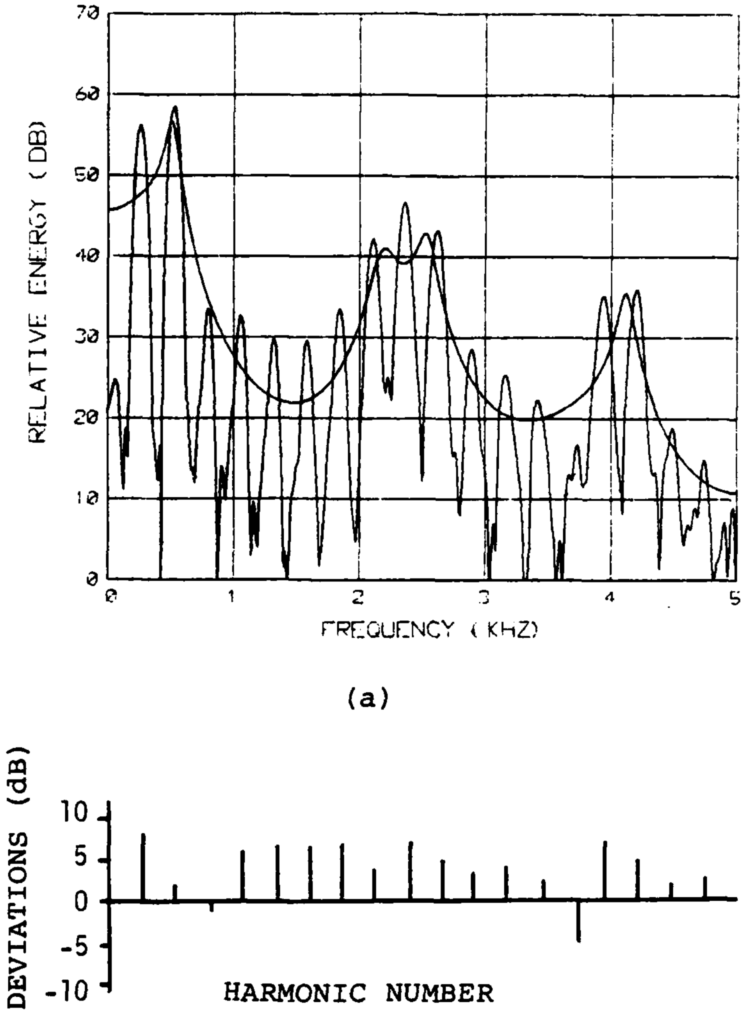
(a)

(b)

Fig. 2. Spectral errors at the harmonics of the fundamental F_0 in LPC modeling, obtained from a male voice (F_0=180 Hz).

(a) Plots of the spectrum of the speech signal (ragged plot) and the spectrum of the corresponding 12-pole LPC filter (smooth plot).

(b) Deviations in dB between the two spectra in (a) plotted as a function of the harmonic number.



(a)

(b)

Fig. 3. Spectral errors at the harmonics of the fundamental F_0 in LPC modeling, obtained from a female voice (F_0=280 Hz)

(a) Plots of the spectrum of the speech signal (ragged plot) and the spectrum of the corresponding 12-pole LPC filter (smooth plot).

(b) Deviations in dB between the two spectra in (a) plotted as a function of the harmonic number.

result of a 12-pole LPC analysis using the autocorrelation method over 20 ms of speech sampled at 10 kHz. The ragged plots in Figs. 2(a) and 3(a) are those of the speech spectrum, while the smooth plots correspond to the model spectrum. The peaks of the speech spectrum correspond to the harmonics of the fundamental frequency; these peaks are further apart in Fig. 3, which was obtained from a female voice, than in Fig. 2, which was obtained from a male voice. Although the LPC model offers a better spectral fit at harmonic peaks, there are still appreciable differences between the two spectra at the harmonics. To see this readily, we have plotted these harmonic differences or deviations between the two log spectra in Figs. 2(b) and 3(b).

To reduce the spectral amplitude errors, one might think of increasing the LPC order (i.e., the number of poles) in an attempt to obtain a more accurate representation of the spectral details. However, to get an accurate representation of the harmonic spectral amplitudes, this procedure requires increasing the number of poles to a value that is on the order of the number of samples in a pitch period. Thus, for a male talker, this procedure may require transmitting about 80-100 LPC parameters, which is clearly an inefficient way of utilizing the limited bit-rate resource.

There are at least two reasons for the above spectral amplitude errors. First, the LPC model does not consider, as part of the model, the periodic nature of the glottal excitation. Second, whereas LPC analysis assumes an all-pole model, the vocal tract transfer function can have both poles and zeros for speech sounds such as nasals, nasalized vowels, and fricatives.

We believe that the spectral amplitude errors resulting from the use of the LPC model are responsible for the unnatural quality of LPC speech. These spectral amplitude errors can be reduced by compensating for them through using an excitation signal with a nonflat spectral envelope.

2.2 A Speech Modeling Methodology for Improved Narrowband Speech Transmission

From the discussions given above and from our experience with narrowband speech coders, we formulated a framework or methodology for speech modeling, which is stated below and which provided the general guidelines for the work performed in this project.

The postulates of the proposed methodology for speech signal representation are as follows:

1. For voiced speech, an accurate representation of the spectral amplitudes at the harmonics of the fundamental frequency is necessary for resynthesis of natural-quality speech.

2. For unvoiced speech, a reasonable spectral envelope model such as the one employed in LPC vocoders provides the required spectral accuracy.

3. An accurate representation of the absolute phases of different frequency components in speech is not essential for natural-quality speech resynthesis. It is important, however, to preserve the phase differences (or relative phase) between the harmonics, for voiced sounds. For unvoiced sounds, a random phase spectrum (phase-versus-frequency characteristic) is required.

Notice that the first two postulates deal with the amplitude spectrum of speech, while the third postulate deals with the phase spectrum. Below, we make additional comments and cite experimental results to support the postulates stated above.

The speech signal for a voiced sound is periodic, and hence its spectrum is properly resolved only at the harmonics. Different spectral estimation methods provide different spectral amplitudes at frequencies between the harmonics. For example, consider the synthetic signal generated by exciting an all-pole filter with a periodic sequence of unit pulses. The discrete Fourier transform (DFT) spectrum of one pitch period of that signal and the spectrum of the impulse response of the all-pole filter agree only at the harmonics [6]. The first postulate above maintains that an accurate representation of the spectral amplitudes is necessary only at the harmonics, for a faithful reproduction of the speech signal. It is interesting to point out that in music synthesis even a slight error in the amplitudes of any one of the first few harmonics can be readily perceived by experienced listeners [7].

Considering the third postulate, an interesting result relating to phase was obtained some time ago using a harmonic synthesizer [8]. In that work, the phases of the harmonics were randomly chosen at the outset and were held fixed during synthesis of voiced sounds. It was found that the quality of the resulting speech was more natural than the quality of the speech synthesized using zero phase for all the harmonics. Notice that the relative phases of harmonics employed in [8] were generally nonzero and were held fixed, thus providing a special case of the phase spectrum considered in the third postulate. The work of Monsen and Engebretson on the variation of the glottal wave as a function of the fundamental frequency F_0 is also relevant [9]. They reported that the harmonic phase relations typically remained invariant with respect to changes in F_0. Finally, we wish to point out that whereas a fixed harmonically related phase spectrum may improve the naturalness of the synthesized speech, not all frequency-dependent phase spectra will have the same beneficial effect. For example, it is well known that allpass filtering the excitation signal or the synthesized speech signal imparts a frequency-dependent phase spectrum to speech but does not improve its naturalness.

A recent experimental work by Atal and David serves to demonstrate the validity of the above set of postulates [4]. Their work has shown that having an accurate harmonic amplitude spectrum is perceptually more important than having an accurate phase spectrum. Also, it has pointed out that use of a harmonic phase spectrum derived from a fixed frequency-dependent group delay (or phase derivative) function along with the exact harmonic amplitude spectrum produces high-quality speech that is perceptually close to the natural speech. The fixed group delay function was obtained by averaging over one or more sentences of speech. This latter result is important for speech transmission applications, since no information about phase needs to be transmitted to the receiver.

Although Atal and David's work provides an "existence proof" for the above speech modeling approach, their method is not directly applicable to a narrowband vocoder as it would require an estimated transmission data rate of about 8 to 9.6 kb/s [4]. In our work, we used the above modeling approach as a starting point and made several changes during the course of this project, as reported in later chapters. We developed good-quality 2.4 kb/s coders operating under practical conditions, which include input speech representative of all speech sounds and of a number of male and female speakers, nonideal pitch and voicing data, and synchronous, noisy channels with up to 1% random bit-errors.

If a speech coder performed a pitch-synchronous analysis over successive pitch periods, transmitted unquantized harmonic spectral amplitudes and phases for every pitch period, and resynthesized pitch-synchronously by an inverse Fourier series procedure, then its output speech would be identical to its input speech. However, the transmission data rate, even under quantization of transmission data, would be quite large. For narrowband speech coding, we achieve data-rate reduction by:

- Not transmitting the phase information
- Analyzing less often than once per pitch period, namely, using time-synchronous analysis at a fixed frame rate
- Effective coding of the harmonic spectral amplitudes by transmitting the deviations between the log spectra of the speech signal and the LPC model at a selected set of harmonics of the fundamental frequency.

We call the resulting speech coder a harmonic deviations (HDV) vocoder. A block-diagram description of the HDV coder is given in the next chapter.

3. DESCRIPTION OF THE BASIC HDV CODER

In this chapter, we describe the basic HDV coder we used in our initial investigations and discuss the databases of input speech we used for testing the HDV coders.

3.1 Block Diagram

Figure 4 shows a block diagram of the basic HDV coder. At the transmitter shown in Fig.4(a), the sampled input speech is analyzed time-synchronously. The LPC analysis (top branch in the figure) consists of removing the short-term dc bias (over the analysis interval) from the speech samples, Hamming windowing, and using the autocorrelation method of linear prediction [5], to extract from the speech signal its energy (or mean-squared value) and p log area ratios (LARs), where p is the order of the all-pole model. For extracting the voicing status and pitch (bottom branch in the figure), we employ the AMDF-DYPTRACK algorithm used in the LPC-10 coder. Section 3.2 describes the modifications to this algorithm that we had to make, to be able to use it under the conditions of our simulation system. The middle branch in Fig.4(a) shows the extraction of spectral deviations, using the input speech, the pitch, and the quantized LPC parameters. In the basic HDV coder, the deviations between

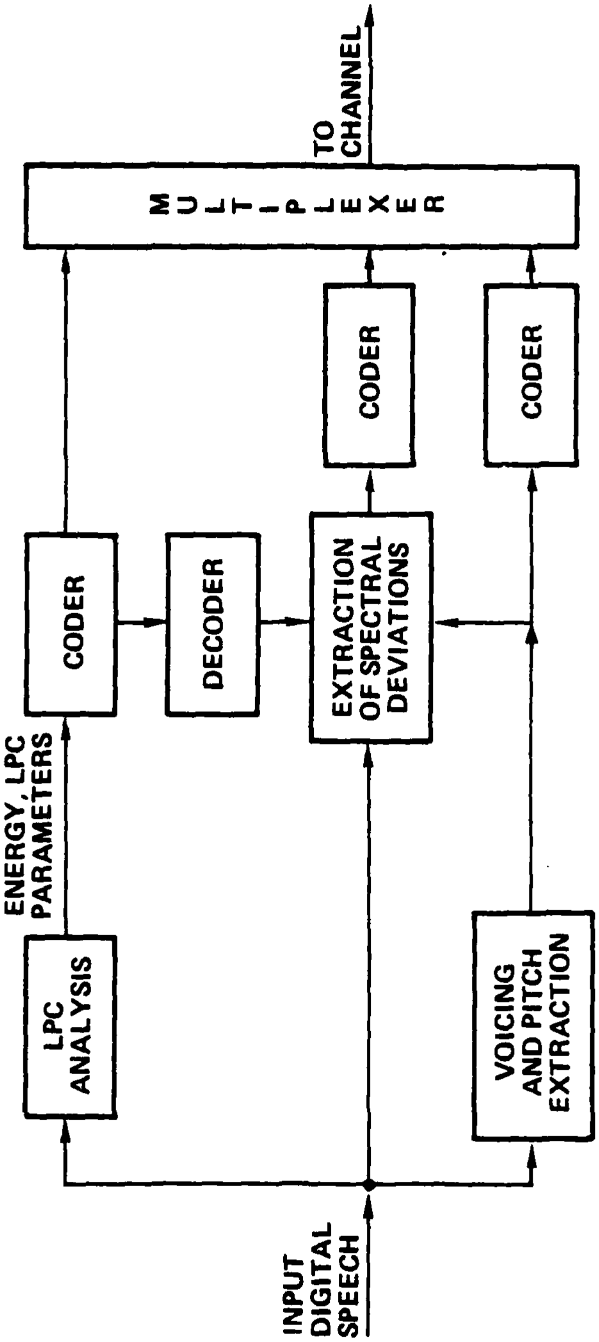


Fig. 4(a). The transmitter of the basic HDV coder.

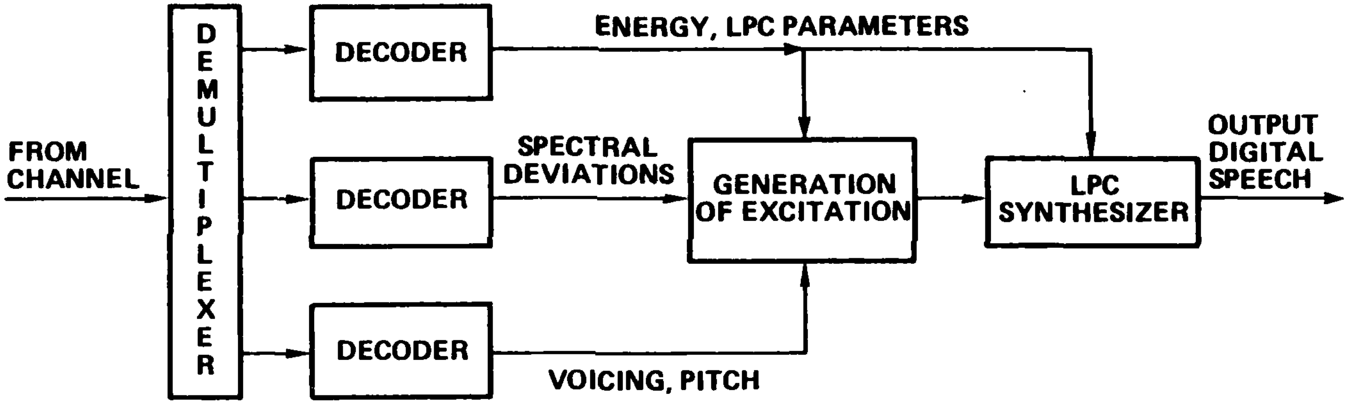


Fig. 4(b). The receiver of the basic HDV coder.

the speech log spectrum and the all-pole model log spectrum are extracted at the harmonics of the fundamental frequency for voiced speech; no spectral deviations are extracted for unvoiced speech. The encoded and transmitted quantities include energy, LARs, pitch, and spectral deviations. Also, we transmit every frame 1 voicing bit and 1 synchronization bit.

At the receiver shown in Fig. 4(b), the decoded harmonic deviations for voiced speech are used to compute a pitch period of the excitation signal for the LPC synthesizer as follows:

- Set the untransmitted deviations to 0 dB
- Generate the linear amplitude spectrum of the excitation signal from the harmonic deviations through exponentiation
- Compute the real and imaginary parts of the DFT of a pitch period of the excitation signal using the above amplitude spectrum and a zero phase
- Perform an inverse DFT.

The pitch-period-long excitation signal is applied to the LPC synthesizer as many times as is required to synthesize one frame of speech. For unvoiced frames, a random noise sequence is used as the synthesizer excitation. We use a gain scaling procedure given in [11] to make the energy of the synthesized speech equal to the transmitted energy of the input speech.

The basic HDV coder uses input speech lowpass filtered at 5 kHz and sampled at 10 kHz, performs 12-pole LPC analysis over 20 ms or 200 samples of speech, and extracts all the transmitted data time-synchronously at a rate of 50 frames/s or once every 20 ms. Since the analysis interval and the frame size (which is determined by the analysis frame rate) are equal for the basic HDV coder, the adjacent analysis intervals do not overlap each other.

3.2 Extraction of Pitch and Voicing

For the extraction of pitch and voicing, as mentioned above, we used the AMDF-DYPTRACK algorithm used in LPC-10 [12]. This algorithm computes an estimate of the pitch for a frame by locating the minimum of the so-called average magnitude difference function (AMDF) and uses a dynamic-programming-based tracking (DYPTRACK) method to refine and smooth the computed pitch estimates. The pitch extracted by this algorithm can take on one of sixty values only, and thus it can be transmitted without further quantization, using 6 bits. To carry out the smoothing just mentioned, the algorithm requires two frames of delay.

The AMDF-DYPTRACK algorithm as implemented in the LPC-10 coder has been "hard-wired" to operate under the conditions of LPC-10 such as 8 kHz sampling rate and 22.5 ms frame size. Values of thresholds, such as the zero-crossing-rate threshold, have been implemented as fixed constants rather than as variables whose values are determined based on vocoder parameters such as frame size. Also, some of the decision parameters used in the algorithm are adaptive in that their values evolve continuously in time. Thus, the algorithm will not, in general, produce satisfactory results if one uses it on a one-sentence-at-a-time basis rather than for continuous speech processing. To resolve these problems and to be able to use the algorithm for different frame sizes and two sampling rates (8 and 1Ø kHz), we made several modifications to the algorithm. Before we describe these modifications, we provide a brief review of the algorithm.

The AMDF-DYPTRACK algorithm contains three distinct sections: a preprocessing section, a pitch extraction section, and a voicing decision section. In the preprocessing section, the current frame of input speech is lowpass filtered at about 1 kHz. The filtered signal is spectrally flattened by passing it through a second-order LPC inverse filter. Both of these signals are stored in buffers for further processing by the pitch extraction and voicing decision sections. In the pitch extraction section, the AMDF function is computed for the inverse-filtered signal. Using the AMDF function, a dynamic-programming-based technique computes a pitch value for the current frame. This technique includes provisions to ensure a smooth variation of the computed pitch values. It should be noted that a pitch value is computed for each frame irrespective of the voicing status of that frame. In the voicing decision section of the algorithm, two voicing decisions are made each frame, one for the first half of the frame and the other for the second half. If a change in the voicing status occurs within a frame, the two half-frame voicing decisions will be different; such a frame is defined as a transition frame. Each voicing decision is determined by the following quantities: an energy measure, a zero-crossing count, previous voicing decisions, and the minimum value of the AMDF function. The zero-crossing count and the energy measure are computed from the data of the previous frame as well as the present frame. Also determined within the voicing decision section is an adaptive parameter (IALO) used to represent the background noise level.

As described above, the algorithm extracts a pitch value and two half-frame voicing decisions each frame. The basic HDV coder accepts only one voicing decision per frame. Based on listening tests and waveform examination of synthesized speech, we chose to declare the voiced-unvoiced transition frame as unvoiced and the unvoiced-voiced transition frame as voiced.

We identified five buffers in the algorithm that require initialization prior to the operation of the coder. Two of these buffers (LPBUF, IVBUF) reside in the preprocessing section of the algorithm. They contain the lowpass- and inverse-filtered data. The other three buffers (IPPATH, IPSTOR, IPSAV) are used in the pitch extraction section of the algorithm. They store data used for smoothing of the pitch values, such as the AMDF function of previously processed frames. We found that by zeroing these five buffers prior to the operation of the coder, satisfactory pitch and voicing decisions were obtained for one-sentence inputs.

Three adaptive parameters were also identified as requiring proper initialization. One of these is a confidence level measure (ALPHAX) used in the pitch extraction section of the algorithm. The other two parameters are a background-noise energy measure (IALO) mentioned above and an energy threshold (IAVENG) used to compute clamps or limits on IALO. To determine the initial conditions of these parameters, they were allowed to evolve in time during the continuous processing of 12 utterances (a total of about 20 seconds of speech) at a frame size of 20 ms. A running average of each parameter was computed each time it was updated. These averages were printed periodically so we could examine how the parameters were changing in time. We found that these averages each approached a relatively constant value after the processing of approximately eight to ten utterances. Tests showed that the "converged" average values provided good initial conditions for the three parameters. Since the values of the parameters are dependent on the number of samples in a frame, the above process was repeated for two other frame sizes: 10 and 30 ms. A linear relationship between each of the initial conditions and the frame size was also obtained.

We found that using a fixed number of magnitude differences for the AMDF computation (regardless of frame size and sampling rate) produced satisfactory pitch and voicing data for each of the three frame sizes (10, 20, and 30 ms) and for each of the two sampling rates (8 and 10 kHz) that we investigated. However, we had to make changes in the program to ensure that adequate speech data is input to the algorithm so that a fixed number of magnitude differences can be properly computed. The revised program employs a pitch analysis interval equal to the frame size or 22.5 ms, whichever is larger.

We identified three (nonadaptive) parameters whose values must depend on the frame size. Two of these are the upper (IHIZC) and lower (ILOZC) thresholds of the zero-crossing count. They were made to vary linearly as a function of the frame size. The third parameter (WT) is a weight used in the computation of the confidence level measure ALPHAX previously mentioned. In the LPC-10 pitch extraction program, this weight is fixed at 1/2. Improvements in the pitch trajectory at frame sizes other than 22.5 ms were observed when this weight was determined by the following relation: $WT = 0.5(22.5/FS)^{1.25}$, where FS = frame size in ms.

The modifications described above were incorporated into the AMDF-DYPTRACK algorithm. From listening tests of synthesized speech and visual observations of the pitch and voicing decisions, we confirmed that the modified algorithm provided satisfactory results.

3.3 Speech Databases

During the course of this work, we used four databases of 11-bit linear PCM speech for testing and evaluating HDV coders: a phoneme-specific database, an all-voiced database, an office-noise database, and a CVSD database. Details of these databases are given below.

3.3.1 Phoneme-Specific Database

This database, described in Table 1, contains six sentences developed previously for formal speech quality testing of LPC vocoders [1, 2]. The first four sentences in Table 1 are phoneme-specific, in the sense that each contains all and only the phonemes of a particular type (glides, nasals, fricatives, and stops), together with vowels. The last two sentences are "general" sentences, which contain several consonant clusters and unstressed syllables. The six sentences in the database were recorded in a quiet environment from 3 male (JB, DD, and DK) and

ID	Sentence	Average Fundamental Frequency in Hz
JB1	Why were you away a year, Roy?	119
DD2	Nanny may know my meaning.	134
RS3	His vicious father has seizures.	195
AR4	Which tea-party did Baker go to?	165
JB5	The little blankets lay around on the floor	124
DK6	The trouble with swimming is that you can drown.	97

Table 1. The six sentences of the phoneme-specific database, with the speaker's average fundamental frequency. The underlined words are emphasized.

2 female (AR and RS) speakers. Speaker JB produced sentences 1 and 5, and the other four speakers produced one sentence each. The average fundamental frequency over each of the six sentences is given in Table 1. We note that these five speakers were selected as being representative of a population of 20 speakers [2]. Because the phoneme-specific database represents a wide range of speech materials and speakers, we have used it most of the time in the testing and comparative evaluation of HDV coders.

3.3.2 All-Voiced Database

This database contains four sentences made up of two all-voiced sentences and 2 male (JS and AH) and 1 female (NC) speakers, with each male producing one of the two sentences and the female producing both sentences. The two sentences used are: 1) "Why were you away a year, Roy?" and 2) "May we all learn a yellow lion roar." The four sentences, recorded in a quiet environment, are identified as JS1, AH2, NC1, and NC2. This database was used primarily in our investigation of pitch-synchronous HDV coders (Chapter 5).

3.3.3 Office-Noise Database

For this database, we digitized six sentences from a sponsor-supplied audio tape recorded in an office-noise environment. Each sentence was produced by a different speaker. We included 3 males and 3 females. This database was used towards the end of this project in evaluating the performance of the optimized coder in the presence of office background noise (Chapter 12).

3.3.4 CVSD Database

The CVSD database has 6 sentences of 16 kb/s CVSD speech, which we digitized from an audio tape provided by the sponsor. The database represents 3 males and 3 females, each producing a different sentence. This database was used towards the end of this project in evaluating the quality of the tandem link between the 16 kb/s CVSD coder and the optimized HDV coder (Chapter 13).

4. EXTRACTION OF SPECTRAL DEVIATIONS

If the pitch period is known exactly, we can compute the harmonic speech spectrum $Q(\omega)$ by computing the DFT of one pitch period of the speech signal and adding the squares of the real and imaginary parts of the DFT. If M is the pitch period in number of samples, then the spectrum $Q(\omega)$ has M values defined at the harmonics $n\omega_0$, $n = 0,1,\ldots,M-1$, where $\omega_0 = 2\pi F_0$, $F_0 = F_s/M$, and F_s = sampling frequency. Because of spectral symmetry, we need to compute $Q(n\omega_0)$ at $n = 0,1,\ldots,[M/2]$ only, where $[x]$ denotes the largest integer not exceeding x. Let us denote the linear prediction all-pole model spectrum as $\hat{P}(\omega)$:

$$\hat{P}(\omega) = \frac{G^2}{|A(e^{j\omega})|^2} = \frac{G^2}{|1+\sum_{k=1}^{p} a_k e^{-j\omega k}|^2}, \qquad (1)$$

where G is the gain of the all-pole filter and a_k, $1 \le k \le p$, are the linear predictor coefficients. We then define the harmonic spectral deviations $dP(n)$, $n=0,1,\ldots,[M/2]$, as

$$dP(n) = 10 \log_{10} Q(n\omega_0) - 10 \log_{10} \hat{P}(n\omega_0). \qquad (2)$$

Since the energy of the synthesized speech is made equal to the energy of the input speech by a gain scaling procedure at the receiver (see Section 3.1), we remove the mean value of the harmonic deviations. From (1) and (2), it can be seen that the zero-mean harmonic deviations can also be computed by inverse-filtering the speech with A(z) to obtain the residual signal, computing the log spectrum of one pitch period of the residual signal, and removing the mean of the log-spectral amplitudes.

The foregoing method of computing the harmonic deviations can be used only if the pitch period is known exactly and if pitch-synchronous analysis is used. For time-synchronous analysis, the pitch value varies over the analysis interval. A single pitch value provided by any pitch extraction method is some sort of an average value for the frame under consideration. Further, the AMDF-DYPTRACK method provides a smoothed pitch value, which is occasionally quite different from the exact, instantaneous pitch. For the conditions of time-synchronous analysis and non-ideal pitch, we investigated three groups of methods for extracting the harmonic deviations for voiced speech. Below, we describe and compare these methods and present several experimental results. Also, we present a method of computing spectral deviations for unvoiced speech.

4.1 Peak Picking

In this method, the spectral deviations at the harmonics of the fundamental frequency are obtained by performing the following steps: compute the log spectrum of the speech signal over the analysis interval (e.g., 20 ms) and of the all-pole LPC filter, locate the peaks of the speech log spectrum, compute the differences between the speech and the LPC log spectra at these peaks, and for each harmonic, compute its desired spectral deviation as a weighted sum of these differences that lie over an interval around that harmonic.

If we denote the power spectra of the speech signal and the LPC model as $P(\omega)$ and $\hat{P}(\omega)$, then the spectral deviation for the nth harmonic is given quantitatively as follows:

$$dP(n\omega_0) = \sum_{i=1}^{m} d_i W_i / \sum_{i=1}^{m} W_i, \tag{3}$$

where

$$d_i = 10 \log_{10} [P(\omega_i)/\hat{P}(\omega_i)], \tag{4}$$

$$W_i = 10^{(X_i - X_{max})/10}, \tag{5}$$

$$X_i = 10 \log_{10} P(\omega_i) - \alpha \left| \frac{n\omega_0 - \omega_i}{\omega_0} \right| , \tag{6}$$

$$X_{max} = \max_i X_i . \tag{7}$$

We note that m in (3) is the number of speech spectral peaks located in the interval $(n\omega_0 - \beta F_0, n\omega_0 + \beta F_0)$ at frequencies ω_i,

$i = 1,2,\ldots,m$. We computed the spectra $P(\omega)$ and $\hat{P}(\omega)$ using 512-point FFT, and we chose the values of the constants α [see (6)] and β empirically. With the weighting function W_i defined in (5)-(7), the larger the peak of the speech spectrum, the bigger the weight, and the nearer the peak to the harmonic in question, the bigger the weight.

One may be tempted to suggest the use of the LPC residual signal spectrum for the above peak-picking procedure, since the resonant information is largely absent in that spectrum and this might ease the problem of locating the peaks. However, in our experience with a similar peak-picking algorithm that we used in the implementation of the mixed-source model [3], we found that the residual-signal spectrum has spurious extra peaks (see Fig. 1(b)), which may lead to erroneous decisions.

4.2 Spectrum Averaging

In this method, pitch-period spectra of the residual signal are computed via DFT over individual pitch periods within the analysis frame, and the spectral deviation at a given harmonic is computed as the geometric mean of the amplitudes of this harmonic for the different pitch-period spectra. The effectiveness of this method depends upon the accuracy with which individual pitch periods are located. With the AMDF-DYPTRACK algorithm, we locate the pitch periods approximately by using the pitch estimate given by the algorithm and by using a simple heuristic procedure, as follows. First, we compute a "refined" estimate of the pitch value for the frame by searching for a peak of the autocorrelation function of the residual signal in the vicinity of the average pitch value given by the AMDF-DYPTRACK algorithm. Second, we locate the largest positive sample of the residual signal in the frame. Normally, we declare this sample to be a pitch pulse; however, if another peak of sufficiently large amplitude immediately precedes this sample, we declare this second peak to be the pitch pulse. Third, we locate the remaining pitch pulses in the frame by repeating the above search procedure in small intervals separated from the first pitch pulse by multiples of the average pitch period. The final step is to locate the zero-crossing preceding each pitch pulse and extract the intervals between consecutive zero-crossings as the desired individual pitch periods. The heuristic procedure just described is quite reasonable, but it does not always locate the pitch periods correctly. For synthesis at the receiver, we use the average pitch value given by the AMDF-DYPTRACK algorithm and the harmonic amplitudes given by the geometric-mean values discussed above.

As an extreme case of the spectrum-averaging method, we investigated a method in which the power spectrum of a single pitch-period of the residual is transmitted. The transmitted pitch period is the one that contains the largest residual sample in the frame. In our experiments, we found that transmitting a single pitch-period spectrum produced more perceivable distortions (roughness for male speakers and "tonal" noises for females) in the output speech than did the geometric-mean spectrum-averaging method.

4.3 Smoothing-Sampling

This method computes (via FFT) the spectrum of the residual signal over the frame, smooths it, and samples the smoothed spectrum at the desired set of frequencies. We investigated several methods of spectral smoothing including linear prediction, cepstral smoothing, and a filter-bank method. Also, we investigated two methods of sampling: F_0 sampling and fixed-frequency sampling. Below, we discuss briefly the three smoothing methods and the two sampling methods.

Linear Predictive Smoothing:

In this method, a smooth spectral envelope of the residual signal is obtained by a high-order (e.g., 20-pole) linear prediction analysis of the residual and by computing the spectrum of the resulting all-pole filter. The degree of smoothing is controlled by the LPC order employed: the lower the order, the higher the degree of smoothing.

Cepstral Smoothing:

This method computes the cepstrum of the residual signal, multiplies the cepstrum with a window, and obtains the smoothed log spectrum of the residual by an inverse FFT. We used both a rectangular window $w_r(q)$ and a cosine-taper window $w_c(q)$ [13], defined below:

$$w_r(q) = \begin{cases} 1 & \text{if } |q|<X \\ 0 & \text{if } |q|\geq X \end{cases} \quad (8)$$

$$w_c(q) = \begin{cases} 1 & \text{if } |q|<X1 \\ \frac{1}{2}\{1+\cos[\pi(q-X1)/(X2-X1)]\} & \text{if } X1\leq|q|<X2 \\ 0 & \text{if } |q|\geq X2 \end{cases} \quad (9)$$

where q is the cepstral variable quefrency and X, X1, and X2 are parameters of the respective window functions. The amount of smoothing is increased by lowering the values of the window parameters.

Filter-Bank Smoothing:

In this method, the amplitude of the smoothed residual spectrum at a given frequency is computed by averaging the amplitudes of the (unsmoothed) residual spectrum over an interval around that frequency. The amount of smoothing produced by this method is increased by increasing the width of the averaging interval.

F_0 Sampling:

In this approach, we sample the smoothed residual spectrum at the harmonics of the fundamental frequency. The sampled values are expressed in decibels and transmitted to the receiver after removing their mean value. The synthesis procedure used at the receiver is the same as discussed in Section 3.1.

Fixed-Frequency Sampling:

This method samples the smoothed residual spectrum at the multiples of a fixed frequency, e.g., 100 Hz. The sampled amplitudes expressed in decibels are transmitted as spectral deviations after removing their mean value as above. However, note that these are not harmonic deviations. At the receiver, linear interpolation between the transmitted deviations is performed to obtain the deviations at the harmonics, which are required for the synthesis.

We compared experimentally the various smoothing-sampling methods. In our initial work, we used fixed, pitch-independent smoothing, i.e., the parameters of the smoothing method were held constant. For this case, all methods yielded about the same overall speech quality. Subsequently, we investigated pitch-dependent smoothing. In this approach, we made the cepstral window parameters vary inversely with the frame F_0 value and the averaging interval in the filter-bank method vary linearly with F_0. We found that with pitch-dependent smoothing, F_0 sampling produced slightly better speech quality than did fixed-frequency sampling. Also, we preferred the speech quality produced by pitch-dependent smoothing and F_0 sampling over the speech quality from the earlier methods using pitch-independent smoothing. Finally, of all the smoothing-sampling methods we investigated, the method using cepstral smoothing with a pitch-dependent cosine-taper window and F_0 sampling produced the best overall speech quality. For this method, we empirically chose the following window parameters [see (9)]:

$$X1 = 0.85/F_0,\ X2 = 1/F_0. \qquad (10)$$

For all subsequent discussions in this report, we refer to this method simply as the smoothing-sampling method.

The smoothing-sampling method is illustrated in Fig. 5, which shows the spectra of the 12-pole LPC model and the residual signal and the smoothed residual spectrum, all computed over a 20 ms interval of voiced speech. The value of F_0 given by the AMDF-

DYPTRACK method for this frame is about 151.5 Hz (pitch period = 6.6 ms). Thus, the smoothed residual log spectrum is sampled at the multiples of this F_0 value, to obtain the harmonic deviations. For the example shown in Fig. 5, the peaks of the unsmoothed residual spectrum correspond approximately to the harmonics of F_0=151.5 Hz.

For the 20 ms frame considered in the above example, we show in Fig. 6 a plot of the excitation waveform of the HDV coder, which was generated using a subset of 10 harmonic deviations only. We selected this subset using the spectral peak adaptive method described below in Section 7.3. For comparison purposes, we show in Fig. 6 plots of the excitation signal used in the LPC coder and the ideal excitation, namely, the residual signal.

4.4 Further Experimental Results

For the experiments reported in this section, we transmitted all spectral deviations at each frame. In all but one experiment, we did not quantize any of the transmitted quantities. Using any of the foregoing methods of extracting the deviations, we found that the HDV coder produced a noticeable improvement in speech quality over the LPC coder, with more improvement for male speakers than for female speakers. The improvement was in the form of reduced buzziness, less tonal

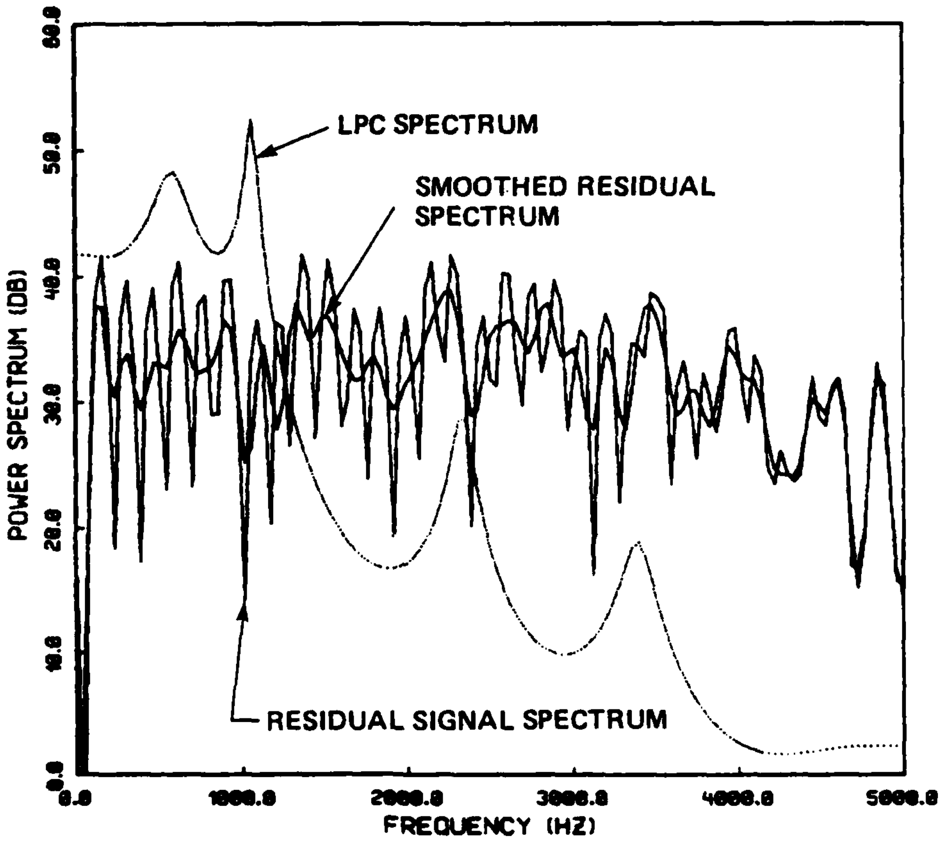


Fig. 5. Extraction of harmonic deviations by the smoothing-sampling method.

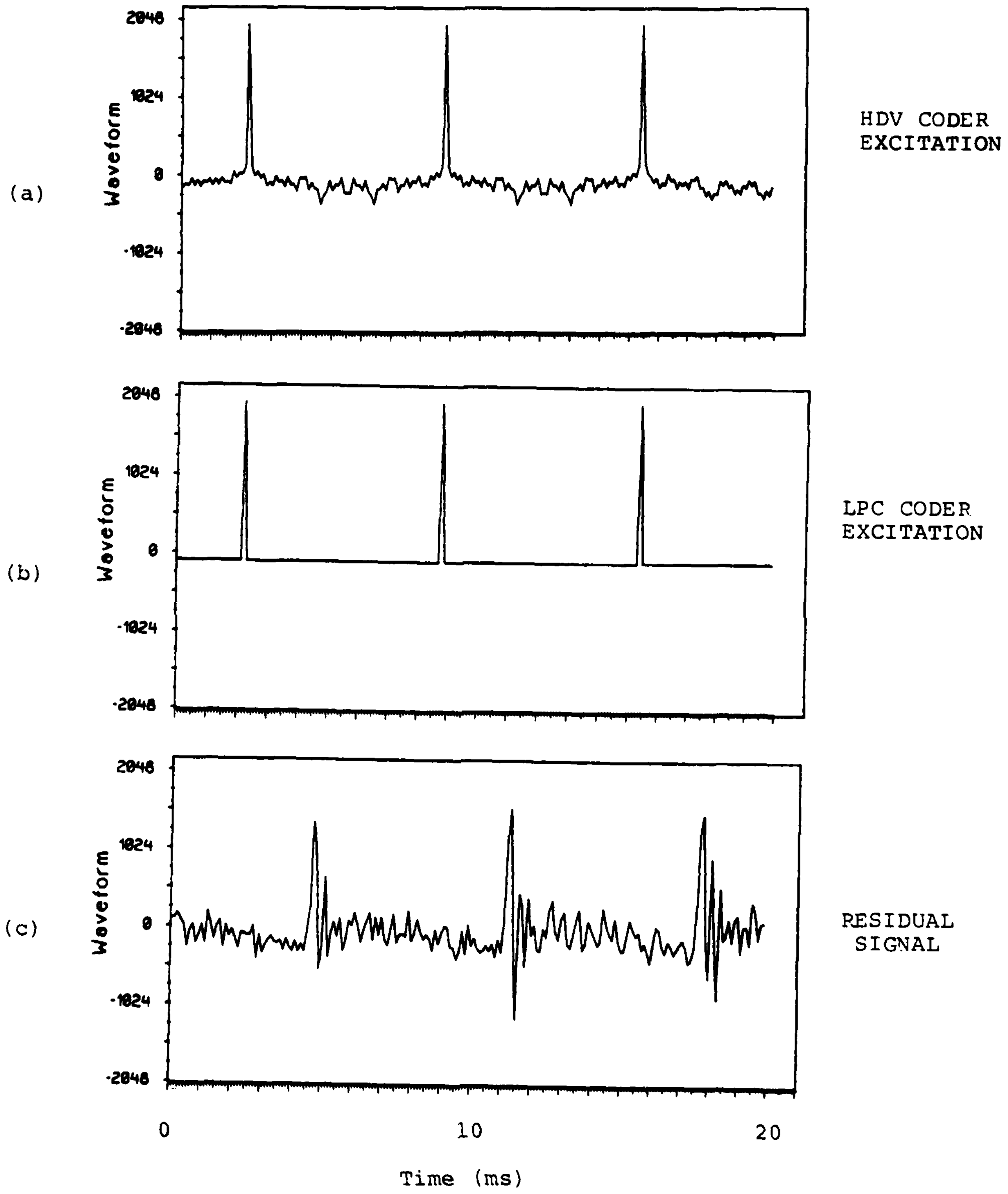


Fig. 6. Comparison of three excitation waveforms:

(a) Zero-phase excitation used in the HDV coder
(b) Periodic pulse sequence used in LPC synthesis
(c) Residual signal (perfect excitation).

noises, and a more natural voice quality. Also, we found that the smoothing-sampling method produced slightly better speech quality than did the other two methods.

Next, we summarize the results of two experiments, which are described in detail in Chapter 8; using these results, we point out a fundamental requirement for obtaining good quality speech from the HDV coder. In the first experiment, we compared HDV coders with LPC order p=8, 10, 12, and 18 poles against a 12-pole LPC coder. For male speakers, 8-pole and 10-pole HDV coders produced "muffled" speech that was not superior to the LPC speech. The 18-pole HDV coder produced only a small speech-quality improvement over the 12-pole HDV coder. In the second experiment, we quantized uniformly the log area ratios of 12-pole LPC and HDV coders. Compared to the LPC coder at 45 bits per frame for LARs, the HDV coder produced better speech quality at 45 bits, about the same quality at 33 bits, and inferior quality at 21 bits. The results of these two experiments indicate clearly that accurate representation of the all-pole spectrum is mandatory for good speech quality. Significant inaccuracies in the all-pole model, caused by either insufficient number of poles or inadequate quantization, cannot be adequately compensated by the addition of the spectral deviations.

We recall that in the basic HDV coder, a pitch-period long excitation signal is computed at the receiver from the transmitted spectral deviations and is applied to the synthesizer repeatedly as many times as required for synthesizing one frame of output speech. The true excitation signal, on the other hand, is the residual signal, which varies from one pitch to the next. We performed an experiment to investigate a condition that lies somewhere between the HDV excitation signal and the residual signal. In this experiment, we extracted one pitch-period long residual signal starting at the zero-crossing before the largest peak over the 20 ms frame and applied it to the synthesizer repeatedly, as in the HDV coder, for speech resynthesis. (Random noise sequence was used for unvoiced frames.) The resulting output speech sounded somewhat more natural than the HDV coder output, but contained perceivable roughness that was not present in the speech from the HDV coder. The roughness problem may be partly due to the imperfect pitch estimate provided by the pitch-extraction algorithm. (This experiment was repeated using hand-corrected pitch data. See Section 5.3 for the results of this case.)

4.5 Spectral Deviations for Unvoiced Speech

In all experiments on HDV coders reported thus far, we used random-noise excitation for unvoiced speech as in LPC vocoders.

In this section we describe a method of using the spectral deviations model for unvoiced speech as well. At the transmitter, spectral deviations are computed from the residual signal using the smoothing-sampling method with cepstral smoothing (cosine-taper window) and 100-Hz sampling. At the receiver, the spectral deviations are used to compute the amplitude spectrum as in the case of voiced speech (see Section 3.1). The amplitude spectrum is combined with a random phase spectrum to produce the real and imaginary parts of the DFT of a 10 ms long excitation signal. An inverse DFT produces the required excitation signal, which is applied as many times as required to synthesize one frame of speech.

In our experiments, we found that the above method produced a slight but perceivable speech-quality improvement over the random-noise excitation. The improvement was in the form of reduced roughness and background noises during unvoiced speech.

4.6 Summary

Of all the methods we investigated for extracting the spectral deviations, the smoothing-sampling method with cepstral smoothing and F_0 sampling for voiced speech and 100-Hz sampling for unvoiced speech produced the best speech quality. This HDV coder produced noticeable improvements in speech quality over the

LPC coder. But, we still observed a substantial difference in the quality between the HDV coder's output speech and the input natural speech. We investigated a number of issues in an effort to narrow this speech quality gap. One set of these issues, reported in Chapter 5, was investigated to examine if ideal conditions of hand-corrected, accurate pitch and voicing data and pitch-synchronous analysis would improve significantly the speech quality of HDV coders. The other set of issues, reported in Chapter 6, was concerned with ways of improving speech synthesis at the receiver.

5. PITCH-SYNCHRONOUS HDV CODER

The pitch-synchronous HDV (PS-HDV) coder, like the basic HDV coder, performs LPC analysis on the speech signal and transmits log area ratios time-synchronously, once every (2Ø ms) frame. In computing the residual signal, the coefficients of the inverse filter are updated each frame at the pitch pulse (see below) closest to the beginning of the frame; at the receiver, the coefficients of the synthesis filter are updated in the same way. The PS-HDV coder, once every pitch period, performs a harmonic analysis of the residual signal using the DFT, computes the energy and the harmonic spectral deviations from the residual signal, and transmits them to the receiver. The excitation signal is generated at the receiver from the harmonic deviations via an inverse DFT, as in the basic HDV coder. The PS-HDV coder FORTRAN program we developed employs an interactive command structure, allowing the user to control various aspects of the coder operation (see Section 5.2) such as what quantities are transmitted and the type of their transmission (time-synchronous or pitch-synchronous). For testing and evaluating PS-HDV coders, we used the four-sentence all-voiced database (see Section 3.3.2) recorded from two males (JS1,AH2) and one female (NC1,NC2). Clearly, the PS-HDV coder requires locating the individual pitch periods in the residual signal. For the all-voiced database, we used the semi-automatic method described next for accurately locating the pitch pulses in the residual signal.

5.1 Extraction of Accurate Instantaneous Pitch

The method we employed involves two stages. In the first stage, we used the so-called data reduction algorithm [14] on the speech signal, to obtain an estimate of the location of the zero-crossing preceding the major positive peak for each pitch period. In this algorithm, two preprocessing steps are performed on the input speech. First, the polarity of the input speech signal is checked and reversed if necessary so that major positive excursions of the waveform represent the onset of pitch periods. Second, the polarity-corrected speech signal is lowpass-filtered with a 32-point FIR filter having a cutoff frequency of 700 Hz. The zero-crossing points are then located by examining the positive excursions of the preprocessed speech signal. Using this algorithm, we processed the four all-voiced sentences and generated disc files containing the pitch markers. We made plots of the speech waveforms with the pitch markers superimposed, to examine the accuracy of the location of the markers. We noted a small number of errors, which we corrected manually.

In the second stage, the hand-corrected speech waveform pitch marks were used to locate the zero-crossings preceding the major peaks of the residual waveform. This step is required because the PS-HDV coder performs harmonic analysis of the residual signal rather than the speech signal. Using the residual signal, an interval of 2 ms centered on each of the speech waveform pitch marks was searched for the largest positive sample of the residual waveform; a residual waveform pitch mark was placed at the zero-crossing immediately preceding each such peak. The latter set of pitch marks was carefully checked for accuracy, hand-corrected, and stored in a disc file for use by the PS-HDV coder.

5.2 Systems Tested

We used the PS-HDV coder program to simulate and test a number of systems, which employed different magnitude conditions, phase conditions, and transmission types. Table 2 summarizes the salient aspects of the systems tested. System TØ, which transmits both harmonic amplitudes and phase pitch-synchronously, is ar identity system in that its output speech must be identical to its input. We used this fact in testing and debugging our implementation of the PS-HDV coder. At the opposite extreme, if no harmonic information is transmitted, the system becomes a pitch-synchronous LPC coder (System T1). Additional descriptions of Systems T2 to T4 and T7 to T11 are given in the next section.

SYSTEM ID	RESIDUAL		TRANSMISSION PS: PITCH SYNCHRONOUS TS: TIME SYNCHRONOUS			COMMENTS
	MAGNITUDE	PHASE	Residual Spectrum	Pitch	Energy	
T0	Original	Original	PS	PS	PS	Identity System
T1	Constant	Zero-Phase	-	PS	PS	PS-LPC Vocoder
T2	Original	Zero-Phase	PS	PS	PS	
T3	Constant	Original	PS	PS	PS	
T4	HDV	Zero-Phase	TS	PS	PS	Smoothing-sampling Method
T5	Constant	Zero-Phase	-	TS	TS	TS-LPC Coder
T6	HDV	Zero-Phase	TS	TS	TS	TS-HDV Coder
T7	HDV	Zero-Phase	TS	PS	TS	
T8	One Pitch period spectrum	Zero-Phase	TS	PS	PS	
T9	One Pitch-Period Residual signal		TS	PS	PS	
T10	HDV	Zero-Phase	TS	PS	PS	Minimum Spectral Error Pitch-period Spectrum
T11	HDV	Zero-Phase	TS	PS	PS	Geometric-mean Spectrum Averaging

Table 2. Description of systems tested in our pitch-synchronous HDV coder study.

We simulated the systems T5 and T6 using the basic (time-synchronous) HDV coder program. System T5 is the LPC coder, and System T6 is the HDV coder using the smoothing-sampling method.

5.3 Experimental Results

We compared the systems in Table 2 via informal listening tests. The PS-LPC coder (System T1) produced the usual speech quality degradations associated with LPC: buzziness, muffled quality, and background tonal noises for female talkers. System T2 differs from System T1 in that it reproduces the residual spectrum exactly. System T2 produced much more natural-sounding speech than System T1 did. In one test, four subjects listened to the triplets of output speech from Systems T1, T2, and TØ in that order, and rated the speech quality of T2 on a scale of 1 to 1Ø, with 1 representing the quality of speech from PS-LPC (T1) and 1Ø representing the quality of the 11-bit PCM speech (TØ). The average rating score was 7.5 for the male speakers (JS1,AH2) and 6 for the female speaker (NC1,NC2). The output speech from System T2 for the female speaker contained some tonal noises.

To determine the relative importance of spectral amplitudes and phase to speech quality, we compared System T2 (original magnitude, zero-phase) with System T3 (original phase, constant magnitude). The speech produced by System T2 was substantially more natural, while the speech produced by System T3 was "noisy." System T3, in fact, produced overall speech quality similar to that of the pitch-synchronous LPC System T1; the speech from T3 was both more natural and more noisy than the speech from T1.

System T4 is a PS-HDV coder in which the harmonic deviations are computed (and transmitted) every 20 ms using the smoothing-sampling method, and pitch and energy are transmitted pitch-synchronously. Since pitch is not constant over the frame, a piecewise linear extension of the transmitted harmonic deviations was used at the receiver to obtain the deviations for individual pitch periods in the frame. The overall speech quality produced by System T4 was *slightly better than midway* between that of Systems T1 and T2. The 10-point rating test mentioned above on the triplet T1, T4, and T2 produced an average score for T4 of 7 for the males and 5 for the female. Some background tonal noise was perceptible at the output of T4 for the female speaker. Systems T1 and T5, pitch-synchronous and time-synchronous LPC coders, respectively, produced roughly similar speech quality, although the speech produced by System T1 was slightly smoother. A more perceptible but still relatively small difference in qualicy was noted in the speech produced by PS-HDV and HDV coders T4 and T6; System T4 sounded more natural. Since these two systems differ in the transmission of both pitch and energy, we simulated System T7 to isolate the source of the perceived speech quality difference. System T7 is identical to T4 except that it transmits energy time-synchronously; no difference was perceived in the speech produced by Systems T4 and T7. Therefore, we conclude that the improvement provided by T4 over T6 is due to the use of accurate instantaneous pitch data. The 10-point rating test on the triplet T5, T6, and T2 produced an average score for T6 of 6 for the males and 4.5 for the female.

System T8 transmits each frame a DFT spectrum obtained from one pitch-period long residual signal. For this purpose, we arbitrarily chose the first pitch period in the frame. Since the residual spectrum transmission is time-synchronous, System T8 is identical to System T4 with the difference that the two systems employ different methods to estimate the harmonic deviations. For male speakers, the speech produced by System T8 was more natural than the speech produced by System T4; for the female speaker, however, System T8 produced louder background tonal noises. System T9, on the other hand, transmits one pitch-period long residual signal and uses it (after truncation for shorter pitch periods or appending with zeros for longer periods) at the receiver rather than using its zero-phase reconstruction as in System T8. Comparing Systems T8 and T9, we found that both produced natural speech of comparable quality for male speakers; for the female speaker, however, System T9 produced roughness while System T8 produced tonal noises. This roughness produced by System T9 was substantially less severe than the roughness produced by a similar system that was simulated using the HDV coder program (see Section 4.4). This reduction in perceived roughness yielded by System T9 is, therefore, due to the use of accurate instantaneous pitch values.

Encouraged by the results from System T8, we investigated several ways of improving its speech quality, especially for the female talker. First, we investigated the effect of a change in the procedure used to generate the deviations for individual pitch periods. Instead of using the piecewise linear extension of the transmitted spectrum, we employed the transmitted harmonic spectral amplitudes directly for all the pitch periods in a frame. Since any pitch period in a frame may be longer or shorter than the first pitch period in that frame, its synthesis requires a larger or smaller number of harmonic amplitudes than the number transmitted. In the modified system, we truncated the transmitted spectrum when a smaller number of harmonic amplitudes was required and extended it with values equal to the average spectral amplitude when a greater number of harmonic amplitudes was required. We found that this modification did not affect the speech quality of System T8. This result leads to the conclusion that the two methods, one preserving the spectral envelope and the other preserving the harmonic amplitudes, yield the same speech quality.

Second, rather than always transmitting the first pitch-period spectrum in a frame as in System T8, we transmitted a spectrum that is in some sense representative of all pitch-period spectra in the frame. We investigated two methods (Systems T1Ø and T11) of determining the representative spectrum. System T1Ø compares each pitch-period spectrum against a corresponding reference spectrum, which is obtained from the spectrum of the residual signal over the frame via the smoothing-sampling method; it transmits the pitch-period spectrum that gives the minimum mean-squared log-spectral difference. System T11 uses the geometric-mean spectrum averaging method (see Section 4.2): it transmits a composite spectrum formed by choosing the spectral amplitude at each harmonic to be the geometric mean of the DFT spectral amplitudes (for the same harmonic) over all pitch periods of the residual signal in the frame. Both Systems T1Ø and T11 produced less tonal noises than System T8 did. We judged the overall speech quality of T11 to be slightly better than that of T1Ø. Comparing System T11 against Systems T1 and T2 and using the previously described rating scale of 1 to 1Ø, listeners rated the speech quality of System T11 as 8.5 for the males and 5.5 for the female talker. (For comparison, the rating scores for T8 were 7.5 and 4, respectively.) Also, System T11 produced noticeable speech quality improvement over Systems T4 and T6.

5.4 Extension to the Time-Synchronous Coder

Of the PS-HDV systems T4, T8, T10, and T11, System T11 produced the best speech quality. However, this system requires the use of accurate instantaneous pitch data, both at the transmitter and at the receiver. The extension of System T11 to the time-synchronous case is the HDV coder using the geometric-mean spectrum-averaging method; this coder, described in Section 4.2, uses a simple heuristic for automatically locating pitch periods within a frame of the residual signal, given the average pitch period for the frame. This time-synchronous system, denoted by T12, produced slightly more natural speech than did System T6 using the smoothing-sampling method, for the all-voiced database. However, as observed in Section 4.4, the opposite was true when we tested the two systems over the six phoneme-specific sentences; System T12 produced worse quality than did System T6. Upon close examination, we found that the AMDF-DYPTRACK algorithm produced more accurate pitch data for the all-voiced (smoothly-varying) sentences than for the phoneme-specific database that has 4 sentences with unvoiced sounds. If one had a pitch extraction method that yields accurate pitch and voicing data, we would recommend the use of the geometric-mean spectrum-averaging method. However, for our application, in view of the robust performance of System T6 under pitch errors, we decided to use the smoothing-sampling method in all our subsequent work.

5.5 Conclusions

Sorting the results presented in this chapter, we make the following conclusions:

1. Pitch-synchronous transmission of the residual signal spectrum (System T2) produces a substantial improvement in perceived speech quality relative to the LPC coder (System T1), both cases using zero-phase reconstruction of the excitation signal. However, there is still a significant quality difference between the speech from T2 and the input speech, especially for high-pitched utterances.

2. Availability of accurate (hand-corrected) instantaneous pitch both at the transmitter and at the receiver produces a perceivable speech quality improvement in several cases we tested, including the HDV coder that uses the smoothing-sampling method. For the case with accurate pitch, the geometric-mean spectrum-averaging method yields the best speech quality.

3. When we use a practical pitch-extraction algorithm, specifically, the AMDF-DYPTRACK algorithm, the smoothing-sampling method leads to better speech quality than does the geometric-mean method. For a robust performance under pitch errors, we therefore recommend the use of the smoothing-sampling method.

4. There is a significant quality difference between the speech from the (time-synchronous) HDV coder T6 using the smoothing-sampling method and the System T2.

In an attempt to reduce the quality difference (1) between the speech from System T2 and the input speech, and (2) between the speech from the HDV coder T6 and the speech from System T2, we investigated a number of methods. These methods and the results are presented in the next chapter.

6. ASPECTS OF SYNTHESIS

We explored a number of aspects of synthesis used in the HDV coder in an effort to improve further the quality of its output speech. Below, we describe these aspects under four major topics and present the results of our experimental work.

6.1 Phase Modeling for the Excitation Signal

Thus far in the report, we have considered zero-phase construction of the excitation signal for voiced speech, at the receiver of the HDV coder. Below, we report on the effect of using a nonzero phase as determined from either frequency-independent or frequency-dependent group-delay functions.

6.1.1 Frequency-Independent Group Delay

In this approach, we circularly shift the zero-phase excitation signal by some amount. A circular shift is a time delay which, in turn, corresponds to a linear phase spectrum. The derivative of the phase, or group delay, is therefore a constant, i.e., independent of frequency. In the first of the two methods we investigated, each pitch-period of the computed zero-phase excitation signal is circularly shifted so that its peak occurs at the same location (relative to the beginning of the pitch period) as the peak of the residual signal [4]. The amount of shift required varies, in general, over individual pitch periods. We applied this procedure to the pitch-synchronous system T2 (see Table 2 and Section 5.3), and transmitted the location of the residual peak pitch-synchronously. However, we found that the output speech from this modified system was no more natural but slightly more rough than the speech from the zero-phase system T2.

Before we discuss the second method, we note that the zero-phase construction of the excitation signal via inverse DFT results in two properties: (1) the first sample is the largest sample in the pitch period and (2) the signal is symmetric around the mid-point (i.e., the second and last samples are equal, etc.). Therefore, large-magnitude samples can occur at the beginning and at the end of a pitch period of excitation signal. Thus, the discontinuities introduced by adjoining such pitch periods occur at places where the excitation signal has large values; this may trigger undesirable transients. The second method tries to alleviate this problem by circularly shifting every pitch period of the excitation signal by half the number of samples in the pitch period. This shifting moves the large-magnitude sample to the middle of the pitch periods (away from the pitch-period boundaries), without changing the spectrum of the excitation signal. When we used this method in the PS-HDV coder T8 (see Section 5.3), we found that the output speech for a female talker contained loud, objectionable roughness. To explain the cause of this distortion, we note that the energy adjustment at the receiver is performed every pitch period via a gain scaling procedure that adds the initial-condition response of the all-pole filter 1/A(z) to the gain-scaled forced response [11]. We observed that placing the large samples in the middle of a pitch period sometimes caused the synthesized speech of the following pitch period to be made up of the initial-condition response only (zero gain scale factor). This problem was solved by performing the energy adjustment once per frame (instead of every pitch period). The resulting output speech, however, was not any better than the speech from the zero-phase system T8.

6.1.2 Frequency-Dependent Group Delay

The idea of using a fixed, frequency-dependent group delay for computing the phase of the excitation signal was suggested by Atal and David [4]. They reported that use of such a phase function in the pitch-synchronous coder T2 (see Table 2 and Section 5.3) produced a significant improvement in speech quality over the zero-phase method.

Group delay is defined as

$$\tau(\omega) = d\phi(\omega)/d\omega, \tag{11}$$

where $\phi(\omega)$ is the phase function. Given a measured group-delay function, we investigated two methods of computing the phase function. Let τ_k and ϕ_k denote, respectively, the group delay and phase at the kth harmonic of the fundamental frequency $F_\emptyset$. The two methods of computing ϕ_k from τ_k are as follows:

$$\text{(Method 1)} \quad \phi_k = \phi_{k-1} + 2\pi F_\emptyset \tau_k. \tag{12}$$

$$\text{(Method 2)} \quad \phi_k = \phi_{k-1} + 2\pi \tau_k. \tag{13}$$

Method 1, which is a discrete approximation of (11), was used in [4]. The harmonic phase for Method 1 varies as a function of $F_\emptyset$. In Method 2 the relative harmonic phase differences are not dependent upon $F_\emptyset$. This property was observed to be a dominant feature of real glottal waves in reference [9].

To measure the group delay of the residual signal as a function of the harmonic number, we performed pitch-synchronous linear prediction analysis over the all-voiced database using the hand-corrected instantaneous pitch data. Every pitch period, we computed the group delay using the following formula, which can be derived using (11) and properties of the Fourier transform:

$$\tau(\omega) = \mathrm{Re}[Y(\omega)/X(\omega)], \tag{14}$$

where Re denotes real value; $X(\omega)$ is the DFT of one pitch period (say, N samples) of the residual signal $x(n)$, $\emptyset \leq n \leq N-1$; and $Y(\omega)$ is the DFT of the modified signal $nx(n)$, $\emptyset \leq n \leq N-1$. We then computed the median group delay of each harmonic over all pitch periods of one or more sentences and stored them on a disc file for later use by the HDV coder. We note that computing the group delay using the formula (14) instead of a discrete approximation of (11) avoids problems due to phase wrapping (or abrupt phase change between $-\pi$ and $+\pi$). Figure 7 shows the median group-delay function we obtained for the sentence JS1. Also shown in the figure is the group-delay function given in reference [4]. The two functions exhibit very similar trends.

For the pitch-synchronous coder T2, we used a fixed, frequency-dependent group delay function and obtained a nonzero-phase excitation signal. For the group delay function, we used the function computed over the same sentence as was being vocoded or the function computed over a different sentence or the Atal and David's function shown in Fig. 7. We used either Method 1 or Method 2 for generating the phase function from the group delay. Disappointingly, none of these cases produced any perceivable change in the speech quality of the coder T2. We have discussed

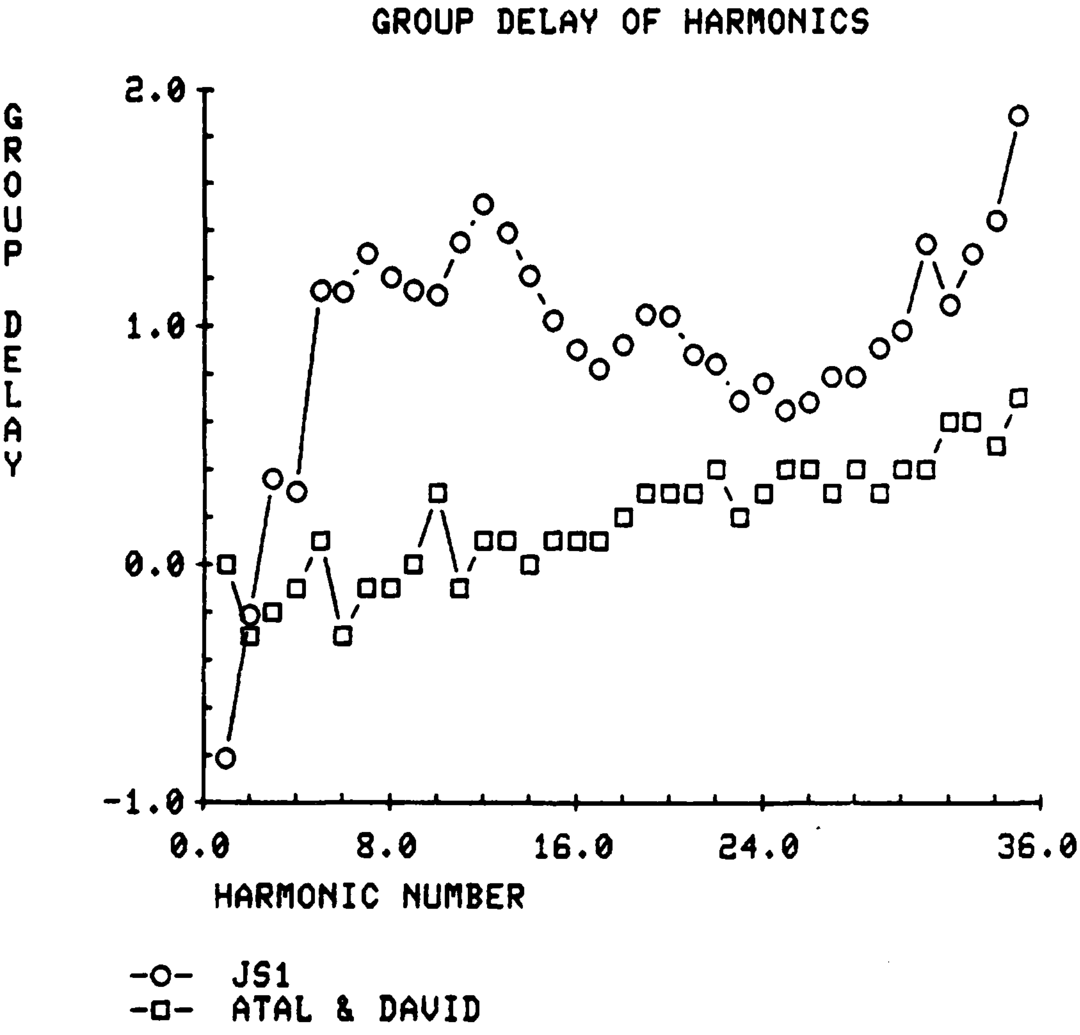


Fig. 7. Two examples of the frequency-dependent group delay function: (1) function we computed for the all-voiced sentence JS1, and (2) function reported by Atal and David [4].

this matter with B. S. Atal. Upon his suggestion, we used the modified covariance method instead of the autocorrelation method for LPC analysis [4] and employed an analysis and transmission rate of 100 frames/s. However, using the group-delay function still did not produce any speech-quality improvement. The difference between our experience and the one reported in [4] may perhaps be due partly to the differences in the input speech databases used in the two studies.

6.2 Mixed-Source Excitation

The mixed-source model allows for simultaneous voiced and unvoiced excitations [3]. The model divides the spectrum into a low-frequency region and a high-frequency region, with the voiced source exciting the low region and the unvoiced source exciting the high region. The cutoff frequency F_C that separates the two regions is adaptively varied in accordance with the changing speech signal [3]. In pure unvoiced frames ($F_C=0$), the unvoiced source alone is used. The transmitted excitation model parameters are F_C and pitch if $F_C>0$; thus, F_C is transmitted instead of the voicing decision used in the voiced/unvoiced source model.

In the time domain, the mixed-source excitation signal is obtained by passing the voiced excitation through a lowpass filter with cutoff F_c and the unvoiced excitation through a highpass filter with the same cutoff frequency F_c, and adding the outputs of the two filters. From the transmitted spectral deviations, the voiced and unvoiced excitations are computed via inverse DFT using, respectively, deterministic and random phases, as explained in previous chapters.

In our work, however, we used a simpler method that "mixes" the voiced and unvoiced excitations in the frequency domain. The harmonic amplitude spectrum is computed as before. The phase spectrum for the harmonics below the cutoff frequency is either set to zero (zero-phase) or generated from the group-delay function as explained in Section 6.1.2. For frequencies above F_c, random phases are used. An inverse DFT is the final step in producing the desired excitation signal.

In our experiments, we found that using the mixed-source model in the HDV coder produced only a slight speech-quality improvement. The primary reason for this result is that use of the harmonic deviations already reduces substantially the buzziness perceived in the LPC speech and yields a certain "fullness" in the perceived speech quality--the two aspects the mixed-source model is known to improve [3]. We feel that the slight speech-quality improvement due to the mixed-source model is not worth the complexity (especially in the analysis) it adds to the HDV coder.

6.3 Frame-Oriented Synthesis

The term frame-oriented synthesis is used to indicate the main difference between this approach and the pitch-synchronous synthesis approach we have considered thus far. The excitation signal is generated one frame at a time in the frame-oriented approach, rather than one pitch period at a time as in the pitch-synchronous approach. The motivation behind the frame-oriented approach is to match the synthesis to the analysis. In other words, spectral deviations are not only computed over a frame but also used to correct the output spectrum over the same time interval.

In the analysis section, we compute, for one frame of speech, the all-pole LPC spectral model, an energy value, an average pitch value, and a set of spectral deviations. At synthesis, we first set the traditional LPC pulse/noise excitation in time for the entire frame, without synthesizing speech. For example, the frame-long excitation waveform may consist of a sequence of pitch pulses separated by zeros; this sequence is constructed as in an LPC receiver, taking into account the continuity of pitch periods across frame boundaries. We compute via FFT the spectrum of the excitation signal, apply to it spectral correction using the transmitted deviations, recombine the magnitude with the phase, and perform an inverse

FFT to produce a frame-long excitation signal, which is then applied to the all-pole synthesis filter. Next, we describe how the spectral correction terms are obtained and used.

During the analysis of a frame, we compute the residual signal by inverse-filtering. We then compute the DFT (via the FFT algorithm) of the residual, which we separate into its magnitude and phase components. Similarly, we compute the DFT of the LPC excitation over a frame and also represent it in terms of its magnitude and phase components. Following these DFT computations, we perform two steps: (1) frequency synchronization of the two spectra and (2) computation of spectral deviations. Frequency synchronization is motivated by the observation that the harmonics of the residual spectrum, due to the quasi-periodic nature of speech, are not always equally spaced over the whole frequency range. To perform frequency synchronization, we partition the spectrum of the LPC excitation into 5 bands, of width 1 kHz each. Each band is cross-correlated with the corresponding band of the residual spectrum such that a best match is attained. The best match is achieved by shifting the LPC band by a certain number of frequency points. The shifts required for individual bands are transmitted to the receiver. The complete spectrum of the excitation signal is reconstructed at the receiver by "cutting, shifting, and splicing" together the bands of the DFT of the LPC excitation spectrum. This excitation spectrum still has a flat spectral envelope, but its harmonic peaks are aligned with those of the residual spectrum. Following frequency synchronization, spectral deviations are computed using any of the methods described in Chapter 4. At the receiver, linear interpolation between the transmitted spectral deviations provides the spectral corrections required at all the FFT frequency points.

To determine the effectiveness of frequency synchronization and spectral correction, we performed a first set of experiments in which we replaced the phase of the DFT of the LPC excitation by the phase of the residual. Under this condition of "perfect phase," we compared four cases: (1) the magnitude of the DFT of the LPC excitation is unchanged; (2) the DFT magnitude is altered only by frequency synchronization; (3) the DFT magnitude is altered only by spectral correction; and (4) the DFT magnitude is altered by both frequency synchronization and spectral correction. These comparisons showed that both frequency synchronization and spectral correction contributed to the improvement of the speech quality over the cases where neither one is used or only one of them is used.

In the second set of experiments, we did not make use of the phase of the residual. We performed an inverse DFT using the corrected magnitude and the phase of the DFT of the LPC excitation. The output speech quality was found to be quite rough. One possible explanation for the degradation in quality is that, with frequency synchronization on the DFT magnitude, the phase and the magnitude are no longer synchronized with one another. To avoid this problem, the first solution we tried is not to use frequency synchronization on the DFT magnitude. The roughness was almost completely eliminated, but the speech quality thus obtained was somewhat buzzy and inferior to the speech quality obtained with pitch-synchronous synthesis. A second solution we tried is to apply the frequency synchronization technique to both the magnitude and the phase. This requires shifting segments of the phase in frequency, to keep the phase synchronized with the DFT magnitude. Attempts made on either the phase or the derivative of the phase failed to eliminate the roughness. As a third solution, we used the all-voiced database and the accurate, instantaneous pitch data (see Section 5.1). The availability of the exact locations of the pitch pulses ensures proper time synchronization between the LPC excitation and the residual waveform. It was hoped that the spectrum of the pitch pulses, placed in time at the pitch marks, would be synchronized with the residual spectrum, thereby eliminating the need for frequency synchronization. However, our experiments showed that although the method improved the speech quality of the coder over that obtained from a pitch-synchronous

LPC vocoder, it produced speech quality inferior to that from PS-HDV coders reported in Chapter 5. Therefore, we decided to continue the use of pitch-synchronous synthesis in the HDV coder.

6.4 Parameter Interpolation

At the receiver of the (time-synchronous) HDV coders we have considered thus far, one pitch-period (10 ms, for unvoiced frames) of excitation is computed and is used repeatedly for the synthesis of one frame of speech. Said another way, the same transmitted parameters, pitch, energy, log area ratios, and spectral deviations, are used over all pitch periods within the frame. We investigated several methods of interpolating between the parameters of two adjacent frames to be able to update their values over pitch periods. The objective of these methods is to synthesize speech with a smoothly varying spectrum. Pitch, energy in dB, and log area ratios are all interpolated linearly. We investigated two methods of interpolating the spectral deviations: (1) direct linear interpolation and (2) spectral interpolation. In the first method, the spectral deviations (in dB) themselves are interpolated linearly. In the second method, the interpolated deviations are obtained by first linearly interpolating the log total spectrum, which is the sum of the log LPC spectrum and deviations in dB, and then subtracting from it the log LPC spectrum obtained from the interpolated LARs. We found that the spectral interpolation method yielded a slight but perceivable speech quality improvement over direct linear interpolation. In our initial experiments, we used pitch-synchronous interpolation and updated the various parameters every pitch period (10 ms for unvoiced speech). Compared to the speech from the uninterpolated case, the speech from the interpolated case had about the same quality for males, but it was smoother and had less tonal noises for females. For all-voiced speech, however, interpolation causes some "slurring" of the output speech; this distortion may be due to excessive smoothing. To reduce the amount of smoothing, we investigated the following modified interpolation method: Instead of linearly interpolating the parameters for each pitch period in a frame, we perform one interpolation only, corresponding to the center of the frame. In generating the excitation signal, we use the transmitted frame data for the pitch periods in the first half of the frame and the interpolated data for the pitch periods in the second half. We found that the modified interpolation method reduced the slurring while retaining the benefits of parameter interpolation. In another experiment we found that interpolating the pitch values at the receiver did not cause any perceivable change in quality compared to the case where pitch was not interpolated. A reason for this result is that the pitch extracted by the AMDF-DYPTRACK method has already been smoothed. Therefore, for computational simplicity, we decided not to interpolate pitch.

6.5 Summary

We reported above our investigation of four groups of methods for improving the synthesis used in the HDV coder: nonzero-phase reconstruction of the excitation signal, mixed-source excitation, frame-at-a-time generation of the excitation signal, and interpolation to update parameters over individual pitch periods. Except for parameter interpolation, all other methods we investigated failed to produce any improvement in the speech quality of HDV coders. The improvement produced by the use of parameter interpolation was mainly in the form of a reduction in the level of the perceived background noises in the output speech.

7. CODING OF SPECTRAL DEVIATIONS

In this chapter we describe our work on coding the spectral deviations. In Section 7.1, we report the measured statistics of the spectral deviations. In Section 7.2, we present several methods of selecting a subset of the deviations for transmission. Section 7.3 contains the results of experiments we performed to compare the different approaches for selecting and quantizing the deviations. During the performance of most of the work reported in this chapter, we used the spectral deviations for voiced speech only. Results for the coding of spectral deviations for unvoiced speech are given at the end of Section 7.3.

7.1 Collection of Statistics

Collecting statistics is not only a necessary step for the encoding of the spectral deviations but is also a good debugging tool that allowed us to examine closely the behavior of the HDV coder algorithm. For each deviation, we computed the histogram, the mean, the median, the 5- and 95-percentile points, and the variance. Initially, we used the smoothing-sampling technique to obtain the deviations, with filter-bank smoothing and 100-Hz sampling. We observed that female speech yielded deviations with larger variance than did male speech. Subsequently, we used the method of cepstral smoothing with F_0 sampling to obtain the deviations, and the variance differences between male and female speech were reduced. Recall from Chapter 4 that the latter method produces better overall speech quality than the former. We used the latter method in the work reported below.

We found that, to a good approximation, all deviations have a "Log-Rayleigh" probability density function. This is to be expected since the deviations, expressed in dB, are the logarithm of the magnitude of the normalized DFT of the residual signal. Assuming that the DFT of the residual has zero-mean Gaussian real and imaginary parts, then the normalized magnitude DFT has a Rayleigh distribution, and the normalized log-magnitude is Log-Rayleigh distributed. We observed experimentally that all harmonic deviations, with the exception of the first and second, have approximately zero mean and median. The mean and median values are 3.8 and 3.5 dB for the first deviation, and 2.1 and 2.5 dB for the second deviation. A plot of the median value and of the 5- and 95-percentile points for all harmoı ic deviations is given in Fig. 8. For this plot, we used 45-bit uniform LAR quantization and collected statistics over several sentences. Except for the first two deviations, the 5-percentile value of the deviation is about -7.5 dB and the 95-percentile value is about +5.5 dB. For coarser quantization of the LARs, i.e., at 35-bit uniform quantization, the ranges of the deviations

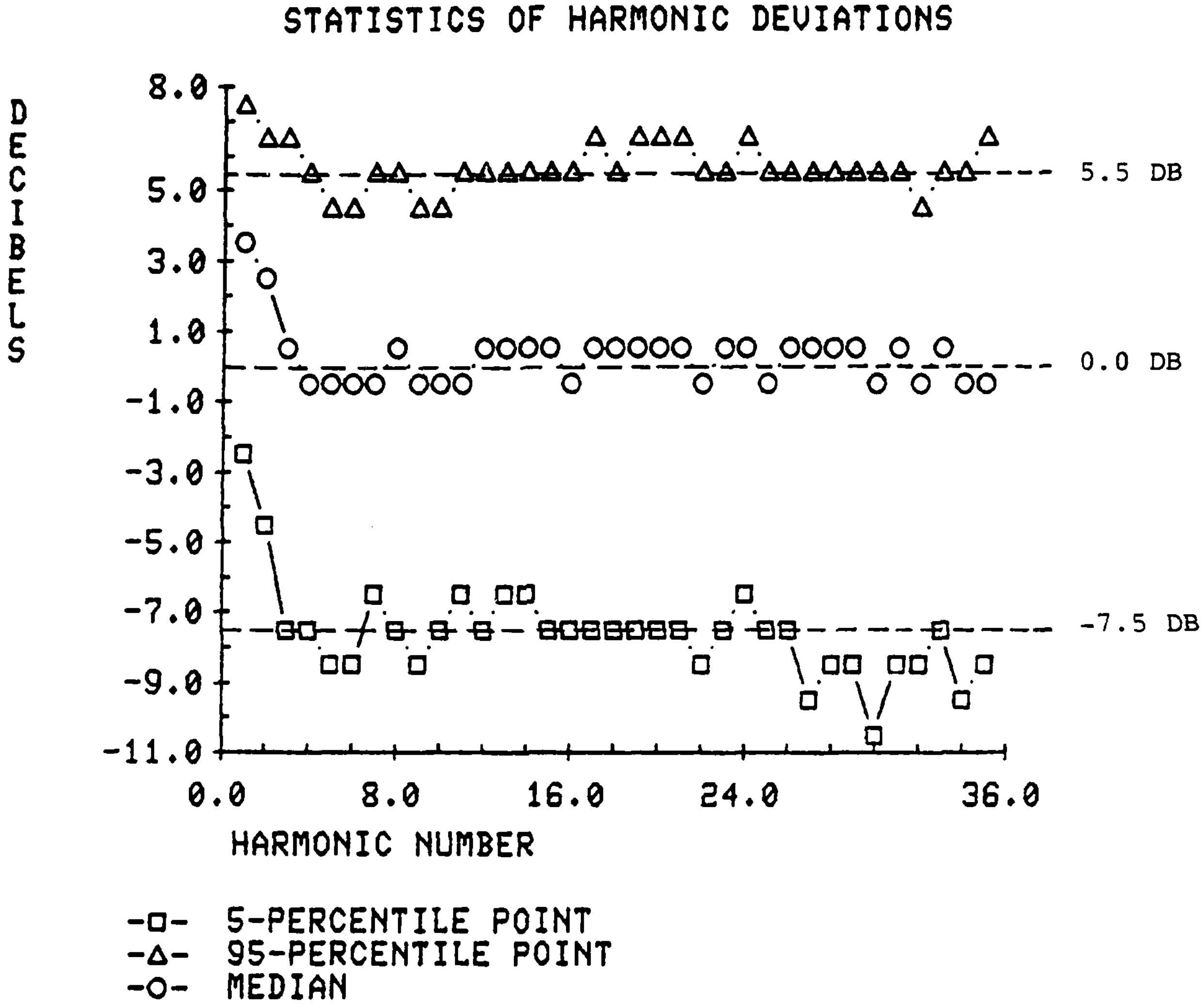


Fig. 8. Three statistics of harmonic deviations: 5-percentile point, 95-percentile point, and median. Each of the three dashed horizontal lines shown in the figure give the approximate value of the corresponding statistic for third and higher harmonics.

increase by about 1 dB, with the 5-percentile value at about -8.0 dB and the 95-percentile value at about +6.0 dB. The actual values of the 5- and 95-percentile points for each deviation were stored on a disc file as the minimum and maximum values, respectively, for later use in the quantization of the deviations (see Section 7.3).

We note here that the positive values we obtained for the mean and median of the first two deviations serve as experimental evidence that the LPC all-pole spectral model, on the average, underestimates the speech spectrum in the vicinity of the first two harmonics. To exploit this result, we implemented a system that uses the following fixed values for the deviations: 3.5 dB for the first deviation, 2.5 dB for the second deviation, and 0.0 dB for all others. The speech from this "fixed" HDV coder sounded more natural and less buzzy than the speech from the LPC coder, although both coders require the same bit rate. We consider this last result as an important one in its own right. We compared this "fixed" HDV coder with an HDV coder that transmits the first two deviations, and found that the latter HDV coder produced more natural-sounding speech than did the "fixed" HDV coder.

7.2 Selective Transmission of Deviations

The number of harmonic deviations per frame is inversely proportional to the pitch of the speaker. Since we use a bandwidth of 5000 Hz, the number of deviations may be as large as 100 for low-pitch male voices and as small as 12 for high-pitch female voices. We investigated several methods of transmitting a selected subset of the deviations in order to limit the information to be encoded. To find a suitable subset of deviations to be transmitted, we examined both static and dynamic bit allocation methods. The work reported in this section was performed without actual quantization of the deviations. Quantization of the deviations is considered in Section 7.3.

In the static or fixed bit allocation approach, we transmit a preselected number of deviations, assuming that each will be encoded using a fixed number of bits. We simulated four cases corresponding to the transmission of (1) the first 10, (2) the first 15, (3) the first 20, and (4) the last 15 deviations. The untransmitted deviations are set to zero (dB) at the receiver. We found that transmitting the first 15 deviations provided better speech quality than transmitting the last 15. Also, we found that the larger the number of transmitted deviations, the better the speech quality. However, for a fixed number of available bits, there is a tradeoff between quantization accuracy and the number of transmitted deviations. This issue is discussed below.

In the dynamic bit allocation scheme, we assumed that 2Ø bits per frame are available for the encoding of the deviations. The available bits are distributed over the deviations in an adaptive fashion that varies from frame to frame. We implemented an integer-bit allocation scheme which, as in Adaptive Transform Coding systems, allocates the bits according to the envelope of the speech spectrum given by the LPC model [15, 16]. Within bit allocation we used a "noise shaping" parameter, g, to control the tradeoff between quantization accuracy and the number of transmitted deviations, i.e., those that are encoded into nonzero bits [16]. We compared two cases: g=Ø.5, where the average number of transmitted deviations was found to be 15, and g=Ø.2, where the average number of transmitted deviations was 18. We found that the case g=Ø.5 provided better speech quality. However, static bit allocation that transmits the first 15 deviations provided better speech quality than dynamic bit allocation with g=Ø.5.

We also investigated two variations of the [illegible] allocation methods. These are discussed in the [illegible] part of our experimental results.

AD-A114 068 BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA F/G 17/2
DEVELOPMENT OF A GOOD-QUALITY SPEECH CODER FOR TRANSMISSION OVE--ETC(U)
MAR 82 V R VISWANATHAN, M BEROUTI, A HIGGINS DCA100-80-C-0039
UNCLASSIFIED BBN-4916 NL

2 OF 2
AD A
114068

END
DATE
FILMED
5-82
DTIC

7.3 Experimental Results

We quantized the deviations using the uniform quantization method. This method divides the range of each deviation into a prespecified number of equal intervals. The range of each deviation is given by the minimum and maximum values, i.e., the 5- and 95-percentile values, discussed in Section 7.1. To evaluate the effectiveness of this uniform quantization technique, we compared three cases: (1) no quantization, (2) 2-bit uniform quantization of all deviations, and (3) 1-bit uniform quantization. The speech outputs for the first two cases were found to be quite similar and often indistinguishable from each other. However, the degradation in going from either case to the 1-bit quantization case was noticeable. Therefore, we concluded that uniform 2-bit quantization of the harmonic deviations is adequate.

Quantizing all the harmonic deviations at 2 bits each requires a large and variable number of bits at each frame. To fix the number of bits to a reasonable value at each frame, we repeated our investigation of the bit allocation schemes described in Section 7.2.

In one experiment, we used 45 bits for uniform quantization of LARs, 6 bits for pitch, 1 bit for voicing, and 5 bits for energy in dB. We simulated three HDV systems that transmit, respectively, the first 10, 15, and 20 harmonic deviations at 2 bits each. All three systems produced essentially the same speech quality, which, in fact, was very similar to the quality obtained when all deviations were transmitted at 2 bits each.

In another experiment, we fixed the number of bits available to encode the deviations at 20 and used 35-bit uniform LAR quantization. For the static bit allocation approach, we compared two cases: encoding the first 10 deviations at 2 bits each versus encoding the first 20 deviations at 1 bit each. We preferred the first case. For the dynamic bit allocation approach, we simulated three cases that correspond to the values of 0.5, 0.35, and 0.2 for the noise shaping parameter g. We preferred the case with g=0.35.

We then investigated two other methods of selecting and transmitting a subset of the deviations. First, we repeated the dynamic bit allocation system (with g=0.35) but using a fixed long-term speech spectrum for bit-allocation rather than the LPC spectrum of each frame. The long-term speech spectral method produced a slight speech-quality improvement over the short-term LPC spectral method, but it was inferior to the case where the first 10 deviations are transmitted at 2 bits each.

Second, we investigated a method that transmits the first two deviations and the two deviations around every peak of the power spectrum of the quantized LPC filter. (The importance of transmitting the first two deviations was demonstrated above in Section 7.2.) To locate the peaks of the LPC spectrum we used a simple 3-point peak-picking procedure. The peaks located by this procedure correspond, in general, to formant peaks. The method transmits, at 2 bits each, the deviations of the two harmonics nearest to the spectral peak--one lower in frequency and one higher. The spectral peak adaptive method just described is illustrated in Fig. 9 for a case where 10 harmonic deviations are transmitted. The short vertical arrows shown in the figure correspond to the transmitted harmonic deviations. Clearly, this method selects the deviations for transmission in an adaptive fashion. It differs from the previously described dynamic bit allocation methods in that all transmitted deviations are quantized using the same number of bits, namely 2 bits. The spectral peak adaptive method proved to be quite effective: with only 6 deviations at 2 bits each the speech quality was found to be the same as for the case where the first 10 deviations are encoded at 2 bits each. We decided to use the spectral peak adaptive method in all our subsequent work.

Considering unvoiced speech, we found that transmitting 6 deviations using the foregoing spectral peak adaptive method produced a slight but perceivable quality improvement over the

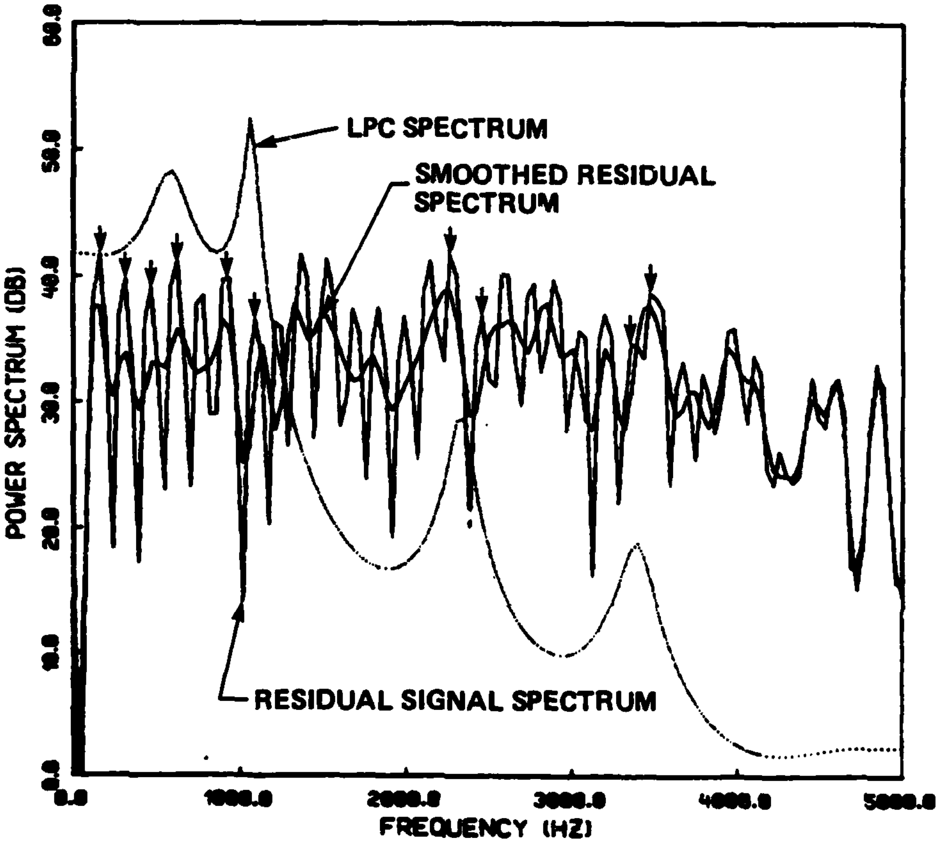


Fig. 9. Illustration of the spectral peak adaptive method for selecting a subset of the harmonic deviations extracted using the smoothing-sampling method. The short vertical arrows in the figure indicate the frequency location of the 10 deviations selected for transmission in this example. (The values of these transmitted deviations are, of course, obtained from the smoothed residual spectrum at the indicated frequency locations.)

case where a random-noise excitation was used. We then investigated a different way of deriving the deviations. In this method, as before, we transmit the first two deviations and the deviations around local peaks of the LPC spectrum. However, the peaks are located by directing the search from high to low frequencies, since important formants of unvoiced sounds are located at the high frequencies. This method, however, did not produce any perceivable improvement over our earlier method in which the search for spectral peaks proceeds from low to high frequencies. Therefore, we decided to continue the use of our earlier approach.

From the results presented above, we conclude that to obtain an improvement in speech quality from the HDV coder over LPC, it is necessary to transmit the first two deviations. Most of the quality improvement obtained by transmitting all the available deviations is preserved by transmitting only the first 10 deviations. We have described a spectral peak adaptive method that selects and transmits a subset of only 6 deviations and achieves essentially the same quality improvement as with transmitting the first 10 deviations. Also, each transmitted deviation can be uniformly quantized using 2 bits, without any perceivable loss in quality. Thus, transmitting the deviations requires an increase in bit rate from 4 to 20 bits per frame, depending on the number of transmitted deviations. The tradeoff between the data rate used for the deviations and the data rate used for LARs or for error protection is discussed below in Chapters 1Ø and 11.

8. QUANTIZATION OF LOG AREA RATIOS

For the design of a 2.4 kb/s HDV coder we are seeking, we must reduce the bit rate required for the transmission of the parameters of the underlying LPC coder to a value that is significantly less than 2.4 kb/s, without causing any loss of speech quality. The bit savings thus achieved can be used to encode the spectral deviations and to provide protection of important parameter data against channel bit-errors. In this chapter, we describe several techniques that we investigated for efficiently encoding the log area ratios. The chapter is organized as follows. In Section 8.1, we discuss the LPC order or the number of LARs to be transmitted. In the next four sections, we describe our work on four quantization methods: uniform scalar quantization, nonuniform scalar quantization, optimal scalar quantization, and two-stage vector-scalar quantization. In Section 8.6, we report on our investigation of fixed and adaptive preemphasis methods.

8.1 LPC Order

We conducted a brief experiment to study the effect of varying the LPC order p (or the number of LARs) on the speech quality of the HDV coder. For this study we did not quantize any of the transmitted data. We considered six values of the LPC order: p=8, 10, 12, 14, 16, and 18 poles. For female speech, the output speech quality improved as we increased p from 8 to 10, but it did not improve further for higher values of p. For male speech, however, we observed a continuous speech quality improvement as we increased p from 8 to 18, with the biggest incremental improvement provided by going from 10 to 12. Both 8- and 10-pole systems yielded a "muffled" quality in the output speech for male speakers. The improvement in speech quality as we increased p beyond 12 was relatively small. This improvement might become even smaller or negligible when the coder parameters are quantized (see Section 8.3 below). Unless stated otherwise, we used p=12 in all our work reported below.

8.2 Uniform Scalar Quantization

To study the effect of quantizing the LPC parameters on the speech quality of the HDV coder, we implemented uniform quantization of the LARs. The lower and upper bounds of each LAR were obtained experimentally by analyzing a large speech database; we computed the bounds separately for voiced and unvoiced speech. Given the total number of bits per frame for LAR quantization, we used a minimum distortion bit allocation method to distribute the bits among the individual LARs. This method, which is described in [2], makes use of mean spectral sensitivities of LARs and produces unequal step sizes for the individual LARs; the method is better than the equal-step-size method reported in [17]. We considered the three bit allocations given below:

- 6,5,4,4,4,4,3,3,3,3,3,3 (45 bits/frame)
- 5,4,3,3,3,3,2,2,2,2,2,2 (33 bits/frame)
- 4,3,2,2,2,2,1,1,1,1,1,1 (21 bits/frame).

For the 45 bits per frame case, the quantization step size varies over the different LARs from Ø.59 dB to 1.69 dB with an average of 1.22 dB, for voiced speech; these values for unvoiced speech are, respectively, Ø.72, 1.37, and 1.Ø7 dB.

We simulated both the LPC and the HDV coder for the above bit allocations. In all cases, energy was quantized logarithmically using 5 bits for a range of 45 dB (minimum=12 dB and maximum=57 dB), and pitch was transmitted using 6 bits. For the HDV coder, all the computed spectral deviations were transmitted without quantization. For each bit allocation, the HDV coder yielded better overall speech quality than did the LPC coder. Compared to the LPC coder at 45 bits per frame for the LARs, the HDV coder produced better speech quality at 45 bits per frame, about the same quality at 33 bits, and inferior quality at 21 bits.

To investigate the tradeoff between the LAR quantization accuracy and the number of transmitted spectral deviations, we compared two fully quantized HDV systems operating at the same total bit-rate of 4900 b/s. In both systems, the spectral deviations were quantized using 2-bit uniform quantization. The first system uses 45-bit LAR quantization and the first 20 deviations, and the second system uses 35-bit LAR quantization and the first 25 deviations. The 45-bit system produced noticeably better quality speech with fewer background noises and less "wobbling" than did the 35-bit system. In fact, a third system with 45-bit LAR quantization and only 10 deviations produced speech of better quality than the 35-bit system, despite its 20 bits per frame lower bit rate. The results of this brief tradeoff study confirm the importance of accurate quantization of the LARs for achieving maximum speech quality. Additional tradeoff experiments are reported in Chapter 10.

8.3 Nonuniform Scalar Quantization

In this method, p nonuniform quantizers are used, one for each LAR. Each quantizer is designed by minimizing the mean-squared quantization error of the corresponding LAR. Minimum mean-squared error (MMSE) quantizers, which were originally proposed by Max [18], are usually derived under the assumption of

Gaussian or other standard probability densities. In our work, we used actual measured densities (histograms) and derived two sets of MMSE quantizers for LARs, one set for voiced speech and another for unvoiced speech. For histogram measurements, we used a "training" database of approximately one minute of read speech from each of 6 male talkers. We allocated the available bits per frame among the individual LARs in such a way that their mean-squared quantization errors were approximately equal. Since we restrict the number of bits per coefficient to be an integer, the mean-squared quantization errors can vary from one coefficient to another, sometimes by a factor of 2 to 4. In practice, bit allocation is a trial-and-error procedure that involves comparing the speech quality resulting from several reasonable bit assignments. The bit allocations that we obtained for four cases, all involving 12 LARs, are given below. The fifth case, listed below for reference, is the 45-bit uniform LAR quantization that we reported in Section 8.2.

- 32 bits: 4 4 3 3 3 3 2 2 2 2 2 2
- 38 bits: 5 5 4 3 3 3 3 3 3 2 2 2
- 41 bits: 5 5 4 4 4 3 3 3 3 3 2 2
- 44 bits: 5 5 4 4 4 4 3 3 3 3 3 3
- 45 bits: 6 5 4 4 4 4 3 3 3 3 3 3

We simulated LPC coders for the five cases, all at a frame rate of 50 frames/s, and compared the output speech for each of the first four cases with that for the fifth case. We found that both 32-bit and 38-bit LPC coders produced considerably worse speech quality, that the 41-bit system produced the same speech quality, and that the 44-bit system produced slightly better speech quality, all compared to the 45-bit uniform quantization case. Similarly, when we added to all systems 6 harmonic deviations, selected by the spectral peak adaptive method described in Section 7.3 and quantized at 2 bits each, the system with 41-bit MMSE LAR quantization produced speech quality equal to that of the 45-bit uniform quantization case.

In Table 3, we give the average mean-squared error (MSE) due to LAR quantization for the two cases: 45-bit uniform quantization and 41-bit nonuniform quantization. (The 38-bit optimal scalar quantizer listed in the table is discussed below in the next section.) The average MSE given in the table was computed over the training database, by adding the mean-squared quantization errors of the 12 LARs and dividing by 12. The error for an individual LAR may differ from the average value by at most a factor of 2. From Table 3, the average MSE of 41-bit nonuniform quantization is only slightly larger than that of 45-bit uniform quantization. Perceptually, as stated above, the two methods produced equivalent speech quality. Therefore, we conclude that for the level of speech quality we are seeking,

Quantizer	Average	MSE
	Voiced	Speech
45-bit Uniform	0.134	0.099
41-bit MMSE	0.145	0.112
38-bit Optimal Scalar	0.138	0.107

Table 3. Average mean-squared LAR quantization error for voiced and unvoiced speech, for three sets of quantizers.

Total Bits per frame	Voiced/Unvoiced	Bit Allocation	Average MSE
41	V	5 5 4 4 4 3 3 3 3 3 2 2	0.098
	U	6 5 4 4 3 3 3 3 3 3 2 2	0.077
38	V	5 4 4 3 4 3 3 3 3 3 2 1	0.138
	U	5 5 4 3 3 3 3 3 3 2 2 2	0.107
35	V	5 4 4 3 3 3 3 3 2 2 2 1	0.186
	U	5 4 4 3 3 3 3 2 2 2 2 2	0.142

Table 4. Bit allocation and average quantization errors, for three sets of optimal scalar quantizers.

MMSE nonuniform LAR quantization saves 4 bits per frame over uniform quantization.

Using nonuniform LAR quantization and the spectral peak adaptive method for selecting the spectral deviations, we conducted an experiment involving a tradeoff between the LPC order and the number of transmitted deviations. In this experiment, we compared three coders, all using a total of 56 bits per frame for LARs and deviations: (1) 12 LARs (44 bits) and 6 deviations, (2) 14 LARs (48 bits) and 4 deviations, and (3) 16 LARs (52 bits) and 2 deviations. We found that compared to coder 1, coders 2 and 3 produced slightly better speech quality for males and more noisy speech for females. The overall speech quality was about the same for all three coders.

8.4 Optimal Scalar Quantization

To reduce further the number of bits required to encode the LARs without incurring any degradation in speech quality, we investigated an optimal scalar quantization method. This method exploits the correlation between the different LARs and uses orthogonal transformation, bit allocation, and MMSE nonuniform scalar quantization. The method is optimal in the sense that it minimizes the expected value of the error

$$E = \sum_{i=1}^{p} [g_i - \hat{g}_i]^2, \tag{15}$$

under the constraint that it uses p scalar quantizers; g_i and $\hat{g}_i$ in (15) are, respectively, the unquantized and quantized values of the ith LAR. The optimal scalar quantization method has been investigated previously by other researchers [19, 15]. The idea of orthogonally transforming the LARs has been used in the design of narrowband LPC vocoders [20, 21].

In the design phase of the optimal scalar quantization method, the transformation matrix, which is composed of the eigenvectors of the LAR data, was used to transform each LAR vector to yield a new vector with uncorrelated components. We separated the analyzed frames into two groups: voiced and unvoiced frames. For each group, we determined the mean LAR vector and the eigenvector matrix, using the same database mentioned above for the design of MMSE quantizers. We transformed all the data to obtain a new database of vectors with zero-mean uncorrelated components. The variances of these components are the eigenvalues of the covariance matrix of the LARs. Using these eigenvalues, we optimally allocated the available bits to minimize the mean-squared quantization error. Following the bit-allocation, we designed a set of MMSE quantizers for the transformed coefficients, and this completes the design phase of this method. For testing, the analysis (coding) and synthesis (decoding) to be performed in the HDV coder are straightforward. A summary of the design, analysis, and synthesis phases is given below:

Design Phase:

- Determine mean LAR vector
- Determine eigenvector matrix
- Transform the data: LARs -> zero-mean, uncorrelated coefficients
- Perform bit allocation
- Design MMSE quantizers

Analysis Phase:

- Determine LAR vector for each frame
- Subtract mean LAR vector
- Transform with eigenvector matrix
- Quantize the transformed coefficients

Synthesis Phase:

- Decode the transformed coefficients
- Perform inverse transformation
- Add mean LAR vector

We implemented the above described procedure and tested it on the 6 phoneme-specific sentences. We emphasize that these test sentences are not part of the training database. Initially, we implemented three cases, which correspond to, respectively, 41, 38, and 35 bits per frame for quantizing the transformed coefficients. The bit allocations for these three cases are given in Table 4 along with the average mean-squared quantization errors. All systems were implemented using 12th order LPC analysis and 6 harmonic deviations coded into 2 bits each as described in Section 7.3. The average MSE for the 38-bit case is also shown in Table 3 for comparison with 45-bit uniform and 41-bit nonuniform quantizers. Upon comparing the average quantization errors and also upon listening to the speech from the various systems, we concluded that the use of orthogonal transformation saves another 3 bits over MMSE quantization. Thus, with the 38-bit optimal scalar quantizers we have accrued a total saving of 7 bits compared to the 45-bit uniform LAR quantization case. Perceptually, the 38-bit optimal scalar quantization case was found to produce speech equivalent to, and in some instances superior to, the speech produced by 45-bit uniform LAR quantization and by 41-bit MMSE quantization. Compared to the 38-bit optimal scalar case, the 35-bit case produced only a slight speech-quality degradation. Finally, we note that the systems with optimal scalar quantization require the storage at both the transmitter and the receiver of two eigenvector matrices (12 x 12 elements each) and of two mean LAR vectors (12 elements each) for voiced and unvoiced speech data.

8.5 Two-Stage Vector-Scalar Quantization

In vector quantization or cluster coding, we minimize the expected value of the error given by (15), without the constraint that forces the use of p scalar quantizers. In this method, the p-vector of LARs is quantized as one entity, using a single quantizer, into one of a finite number of predetermined LAR vectors or templates. Vector quantization has been used for quantizing the LPC coefficients with up to 10 bits [22]. Such low-bit quantizers do not produce the level of quantization accuracy and speech quality that we are seeking. Vector quantization with total bits per frame that is significantly larger than 10 requires very large training databases (typically several hours of speech) for quantizer design, and excessive storage and computation for coding and decoding.

From work performed at BBN as part of another project [23] it has been recently found that vector quantization behaves like the optimal scalar quantization approach at bit rates larger than about 15 bits per frame, and that it saves about 3 to 5 bits over optimal scalar quantization at data rates lower than 15 bits per frame. To exploit this apparent advantage of vector quantization over scalar quantization, while achieving good quantization accuracy, we implemented a two-stage vector-scalar technique. The main motivation behind this study was our desire to decrease the required number of bits still further while achieving the accuracy of, say, the 38-bit optimal scalar quantization discussed in the previous section.

The method consists of a vector quantization stage followed by a scalar quantization stage. In the first stage, the vector $\underline{g}$ of the LARs of a frame is quantized as the jth template $\underline{g}(j)$. We used 1 to 5 bits for this first-stage cluster coding. We implemented the K-means method, which compares the LAR vector $\underline{g}$ with all the templates and selects the template $\underline{g}(j)$ that has a minimum Euclidean distance from $\underline{g}$. With each template, we store a pxp matrix $\underline{A}(j)$ of eigenvectors of the training data of the corresponding cluster. The deviation vector $\underline{d}=\underline{g}-\underline{g}(j)$ is computed and transformed using the eigenvector matrix $\underline{A}(j)$ i.e., $\hat{\underline{d}}=\underline{A}(j)\underline{d}$. The transformed deviation vector $\hat{\underline{d}}$ is quantized in the second stage using MMSE nonuniform scalar quantizers. To design these scalar quantizers, we combined the deviation data of all the clusters, which were computed over the training database; probability histograms were then computed for this combined deviation database. The template index j used in the first stage and the vector of quantization levels obtained in the second stage constitute the transmitted quantities. Decoding consists of scalar decoding of the deviations, inverse transformation of the decoded deviation vector, and addition of the template $\underline{g}(j)$. For example, given a total of 37 bits, one could perform an initial cluster coding stage at 2 bits followed by a second stage with 35-bit MMSE scalar quantization of the deviations. We denote this two-stage quantizer by the pair (2,35).

We designed and implemented the necessary sets of vector and scalar quantizers for voiced and unvoiced speech separately. For a total of 37 bits per frame, we simulated the 4 two-stage quantizers shown in Table 5. The table gives the average mean-squared quantization errors for the 4 cases; these errors were computed over the training database, during the quantizer design phase. From Tables 4 and 5, we see that the two-stage 37-bit quantizers give about the same quantization error as the 38-bit optimal scalar quantization method. This result indicates that the two-stage vector-scalar quantization method provides an apparent saving of one bit over the optimal scalar method. However, on the one hand, the two-stage approach is more complex than the optimal scalar approach because of its storage and computational requirements. And, on the other hand, we found the two-stage approach to be less robust against training-data versus test-data differences than the optimal scalar approach. This latter issue of robustness is discussed next.

When we tested the four 37-bit two-stage cases shown in Table 5 on the 6 phoneme-specific sentences, we found that they produced speech quality roughly equivalent to that of the 35-bit

Two-Stage Quantizer	Bit Allocation for the Second Stage (same for Voiced & Unvoiced cases)	Average MSE	
		Voiced	Unvoiced
(1,36)	4 4 4 3 3 3 3 3 3 2 2 2	0.140	0.118
(2,35)	4 4 3 3 3 3 3 3 3 2 2 2	0.151	0.117
(3,34)	4 4 3 3 3 3 3 3 2 2 2 2	0.151	0.118
(5,32)	4 4 3 3 3 3 3 2 2 2 2 1	0.138	0.117

Table 5. Average mean-squared quantization error for four two-stage vector-scalar quantizers, each using a total of 37 bits per frame.

optimal scalar case and slightly inferior to that of the 38- and 41-bit cases. Upon examining the average quantization errors computed over the phoneme-specific database and given in Table 6, we find that the error produced by any of the two-stage quantizers is about equal to that produced by the 35-bit scalar quantization method. This explains why the two cases produced similar speech quality. It also points to the issue of robustness: there seems to be a loss of 2 bits due to the two-stage approach rather than the anticipated 1-bit saving. To confirm our findings, we designed the (2,35) two-stage quantizer and a 37-bit optimal scalar quantizer using the 6 phoneme-specific sentences as the training database, and tested them over the 4-sentence all-voiced database. The design and test average MSE values were, respectively, 0.124 and 0.270 for two-stage quantization, and 0.121 and 0.185 for optimal scalar quantization. To examine the effect of increasing the size of the first-stage vector quantization, we designed a (10,24) two-stage quantizer over a training database of about 14 minutes of read speech from a number of male speakers. The average MSE obtained during the design phase was 0.154; from this and Table 4, we find that the two-stage quantizer produces an apparent saving of about 3 bits. However, the (10,24) quantizer produced a test error of 0.255 over the phoneme-specific database; this error is, in fact, larger than the corresponding test error

Quantizer		Average MSE
Optimal Scalar	35-bit	0.200
	38-bit	0.154
	41-bit	0.110
Two-Stage	(1,36)	0.236
	(2,35)	0.198
	(3,34)	0.195
	(5,32)	0.197

Table 6. Average mean-squared error over the phoneme-specific test database, for several scalar and vector-scalar quantizers. The average MSE was computed over both voiced and unvoiced speech.

produced by the 35-bit optimal scalar method (see Table 6). Listening tests confirmed that the overall speech quality of the HDV coder over the phoneme-specific database was worse for the two-stage quantizer than for the scalar case. From these results, we conclude that the two-stage quantizer is more sensitive or less robust to training-data versus test-data differences than the optimal scalar quantization approach. Therefore, we decided to use optimal scalar quantization in the final HDV coder design.

8.6 Preemphasis

Preemphasizing the input speech signal reduces its spectral dynamic range, and hence it is generally thought of as beneficial for the encoding of the LARs. We investigated two types of preemphasis with the hope of reducing the number of bits required to encode the LARs without causing any perceivable degradation in the output speech: fixed, first-order preemphasis (Section 8.6.1) and adaptive preemphasis (Section 8.6.2). The adaptive method we investigated is known as differential linear prediction [24].

8.6.1 Fixed Preemphasis

We implemented fixed, first-order preemphasis in two ways: (1) as a preprocessor directly on the input speech (external preemphasis) and (2) as part of parameter coding (internal preemphasis). The inverse process, deemphasis, is applied to the final speech output in case 1 and on the decoded LPC coefficients in case 2. The two types of preemphasis are equivalent for the LPC coder. They are not equivalent, however, for the HDV coder. The residual signal used for computing the harmonic deviations is derived from the preemphasized speech for external preemphasis and from the unpreemphasized speech for internal preemphasis. The preemphasis filter used is the first-order filter given by $(1-0.9z^{-1})$. For the preemphasized case, we used an LPC order of 11 (one less than the value used for the unpreemphasized case).

In our initial investigation of preemphasis, we used uniform quantization of LARs and transmitted all the extracted spectral deviations without quantizing them. When the bits per frame, B, used for LAR quantization is large (e.g., B=42), internal and external preemphasis yielded about the same speech quality at the output of the HDV coder. For lower values of B (e.g., B=35), using external preemphasis caused more roughness in the output speech than using internal preemphasis. Therefore, we used internal preemphasis in our subsequent experiments on preemphasis.

We compared the HDV coder using preemphasis with the coder using no preemphasis. For B=42 bits per frame, the two cases produced about the same speech quality. For B=35, the two cases produced different types of speech quality distortions. The coder with preemphasis occasionally produced roughness in the output speech, while the coder without preemphasis produced lack of clarity in some sentences. There was no clear indication as to the superiority of one system over the other.

In our subsequent investigation of preemphasis, we transmitted 6 deviations at 2 bits each, using the spectral peak adaptive method described in Section 7.3. In one experiment, we used 41-bit MMSE nonuniform scalar quantization of LARs, with the following bit allocation for voiced and unvoiced frames: 5,5,4,4,4,4,3,3,3,3,3. The HDV coder produced slightly worse speech quality for the preemphasized case than for the unpreemphasized case listed in Table 4.

In another experiment, we used preemphasis only for voiced frames, simulated LPC and HDV coders with 35-bit and 41-bit optimal scalar quantization of LARs, and compared them against the corresponding coders without preemphasis. In general, the cases without preemphasis were preferred to the preemphasized cases. Once again we noted that preemphasis tends to cause roughness in the output speech. From the results presented above, we decided not to use fixed, first-order preemphasis in our HDV coder.

8.6.2 Differential Linear Prediction

The differential linear predictive coding (DLPC) method is based on a model of speech that assumes that for voiced sounds, the speech spectrum varies slowly on a frame-to-frame basis [24]. The DLPC algorithm attempts to use the correlation between adjacent frame spectra for efficient coding of the spectral parameters (LARs). The method is a backward-adaptive technique so that no information concerning the adaptation is transmitted. Pitch and gain are transmitted as in the LPC coder.

In DLPC, the spectrum of a voiced frame is estimated as a function of the spectrum of the previous voiced frame. (The first voiced frame at a U-V transition and all unvoiced frames are not encoded differentially.) The DLPC method estimates the all-pole model of a voiced frame as the product of two quantities [1/P(z)] [1/A(z)], where P(z) is derived (as described below) from the overall all-pole model of the preceding voiced frame, and A(z) is computed using the linear prediction method over the signal that is obtained by preemphasizing the input speech signal of the present frame with the filter P(z). In our implementation, we considered the orders of the two filters

1/P(z) and 1/A(z) to be each equal to 12, used the autocorrelation method of linear prediction for computing A(z), performed the foregoing preemphasis operation in the autocorrelation domain (internal preemphasis) instead of on the speech signal, and implemented the synthesis filter as a single 24th order all-pole filter instead of a cascade of a 12th order synthesis filter 1/A(z) and a 12th order deemphasis filter 1/P(z). Notice that the 12th order internal preemphasis mentioned above requires computing the autocorrelations R(Ø),R(1),...,R(24) of the input speech.

The coefficients of the filter P(z) are computed at the transmitter (for preemphasis) and at the receiver (for deemphasis), by using the following two-step procedure. First, compute the optimal 12th order all-pole model for the 24th order all-pole model of the previous frame. This step is performed by computing the reflection coefficients of the 24th order model, retaining only the first 12 reflection coefficients, and computing the predictor coefficients corresponding to these 12 reflection coefficients. Second, the resulting 12 predictor coefficients are multiplied by an exponential "window," i.e.,, the kth coefficient is multiplied by $\exp(-\pi kW)$, where W is a parameter to be optimized. This second step widens the bandwidths of the poles of the filter 1/P(z) by W Hz.

An advantage of this method of estimation of the all-pole model is that the resultant adaptive preemphasis and deemphasis filters are completely specified by previously transmitted information, i.e., the resulting DLPC system is a backward-adaptive system. If it is assumed that the overall spectral distortion is a function of only the quantization accuracy, then the number of bits necessary for coding at any given distortion for LPC and DLPC may be calculated and compared. For each system, the variance of each parameter to be transmitted is calculated. (In this study, these are the variances of the LARs). If the variances of the parameters to be transmitted are decreased, then encoding of the parameters will require fewer bits. The difference in the number of bits necessary to encode the LPC and DLPC parameters at the same distortion is proportional to the difference in dB of the geometric means of the variances of the LPC and DLPC parameters with the proportionality constant being 1/6.02. (Recall the 6 dB per bit rule.)

Using this calculation method, the savings in number of bits for encoding each spectrum using DLPC rather than LPC at equal average distortions was found as a function of the bandwidth parameter W. For this calculation, we used our 6-sentence phoneme-specific database. In estimating the variances of the DLPC parameters, we considered only unquantized parameters.

Since the effect of parameter quantization will be propagated by the adaptive preemphasis, the results given below should be interpreted cautiously. The bit-rate reduction was calculated both with and without eigenvector rotation (orthogonal transformation). The maximum reduction occurred for W=30 Hz, and it was 5.98 bits for the case without eigenvector rotation and 5.78 bits for the case with eigenvector rotation. When W was increased to 600 Hz, the bit savings provided by DLPC over LPC in the above two cases decreased to, respectively, 3.03 bits and 3.47 bits.

In a preliminary test, we listened to the output speech from the unquantized LPC system and from the unquantized DLPC system with the bandwidth parameter W set at 30 Hz, which yielded the maximum bit savings as mentioned above. The speech from the DLPC system was significantly more degraded than the speech from the LPC system. The degradations were in the form of roughness, muffled quality, and "metallic sounds." These degradations decreased as the bandwidth W was increased; at 600 Hz, the speech from the LPC and DLPC systems sounded identical. For W=600 Hz, however, the bit savings due to DLPC decreases to approximately 3 bits per frame, as mentioned above.

To complete our experiments with DLPC, we included quantization of the LPC parameters. Optimal scalar quantizers (employing eigenvector rotation) were designed and implemented at several bit rates. In one experiment, we found that at any given bit rate for the transmission of the LAR parameters, the addition of harmonic deviations improved the quality of the DLPC system. In another experiment, we compared the DLPC systems with W=30 Hz and W=600 Hz. We used 33 bits per frame for LAR quantization. For systems both with and without harmonic deviations, the parameter setting W=600 Hz resulted in higher quality output speech.

The results presented above indicate that although DLPC saves up to 6 bits per frame over LPC from a coding theory viewpoint, it saves only 2 bits per frame when compared on a perceptual basis using listening tests. Due to the additional complexity of the DLPC system and the potential problems in the presence of channel errors, we decided against the use of DLPC in our system.

8.7 Conclusions

In this chapter, we described our extensive work on the efficient encoding of the LPC parameters. We found that fixed, first-order preemphasis does not help in reducing the number of bits required to quantize the LARs without causing speech degradations. As for adaptive preemphasis, we investigated the method called differential linear predictive coding (DLPC). We found that DLPC is not very effective in exploiting the frame-to-frame correlation of speech spectra. In the next chapter we describe a more effective technique, called variable frame rate transmission, to reduce the bit rate by exploiting the frame-to-frame correlation of LARs.

We presented four methods of quantizing the LARs: uniform scalar, MMSE nonuniform scalar, optimal scalar, and two-stage vector-scalar. Of the three scalar methods, the optimal scalar method produces the largest bit savings without causing any speech-quality degradation. The mean-squared quantization error for the 38-bit optimal scalar method is about the same as for the 45-bit uniform scalar method or for the 41-bit MMSE nonuniform scalar method. Further, speech quality produced by 35-bit optimal scalar quantization is only slightly worse than that from 45-bit uniform quantization. The two-stage vector-scalar method provides a potential saving of 1 to 3 bits per frame over the optimal scalar method. In practice, however, the performance of the two-stage method degrades appreciably whenever the input speech is different from the speech used for designing the quantizer; this degradation wipes out the bit savings over the optimal scalar method. Therefore, we decided to use the optimal scalar quantization method in all our subsequent work.

9. VARIABLE FRAME RATE TRANSMISSION

For a fixed frame rate transmission at 5Ø frames/s and a bit-rate of 2.4 kb/s, we have available 48 bits per frame, which is just enough to transmit pitch (6 bits), voicing (1 bit), energy (5 bits), LARs (35 bits), and one synchronization bit. This 2.4 kb/s LPC vocoder has been used in our work as a reference against which to compare HDV coders. Thus, to transmit deviations and provide error protection of some of the transmitted quantities in the noisy channel application, we need to reduce the transmission frame rate below 5Ø frames/s, without lowering the output speech quality. For this purpose, we use variable frame rate (VFR) transmission in which the interval between adjacent parameter transmissions varies to match the changing properties of the speech signal. Methods of VFR transmission and their applicaticns have been investigated in previous DoD-sponsored projects at BBN [25, 2, 26, 27, 28]. The block-encoded VFR scheme described in Section 9.1 provides a fixed bit rate, which is required to transmit over a synchronous channel. This VFR scheme has been used previously in the design of a robust 2.4 kb/s LPC vocoder [26, 27]. In Sections 9.2 to 9.4, we present experimental results on VFR-HDV coders. For the discussions in Sections 9.1 to 9.4, we assume that spectral deviations are transmitted at a fixed frame rate. VFR transmission of the deviations is considered in Section 9.5.

9.1 Block-Encoded VFR Transmission

In VFR transmission, LPC analysis is carried out at a fixed frame rate, but the extracted vocoder parameters (pitch, energy, and LARs) are transmitted only when necessary as determined by appropriate transmission criteria. The VFR method results in a variable duration between successive transmissions and yields a lower average transmission frame rate than the fixed frame rate method (which transmits all the extracted vocoder parameters). At the receiver, parameters for the untransmitted frames are obtained by linearly interpolating between adjacent transmitted frames.

In block-encoded VFR transmission, no transmission decisions are made until a block of speech, say, N frames, has been analyzed. A particular sequence of, say, M frames in the block is then selected for transmission. For a given VFR system, N and M are predetermined and fixed, e.g., $N=7$ and $M=5$. We denote the VFR scheme by the pair (M,N). For a transmitted frame, pitch, energy, and LARs are transmitted together. Since there are $N!/[M!(N-M)!]$ ways of choosing M frames out of N, we transmit a header at the beginning of each block to identify the transmitted frame sequence. Following the header, we transmit the frames of data in their proper time order. This approach guarantees that frame boundaries are always properly identified at the receiver.

Channel bit-errors may cause misinterpretation of the header, which introduces only small, limited time shifts between frames at the receiver.

The particular sequence of M frames chosen for transmission is that which minimizes the total error, computed over the block, between the unquantized parameters on the one hand and the quantized or interpolated parameters on the other hand. An exhaustive search is made over the set of allowable sequences of M frames out of N, and the sequence with minimum total error is chosen. The total error for a given frame sequence is computed as follows. The errors in the parameter values are computed at each frame and summed over the block, independently for each of the three parameter sets to obtain the errors E_p , E_g, and E_c, for pitch, energy (or gain), and LAR coefficients, respectively. The parameter errors over the block are then weighted and added to form the total error for that sequence:

$$ET = w_p\ E_p+w_gE_g+w_cE_c. \tag{16}$$

The weights are chosen empirically to obtain a proper mix of the three types of errors and achieve a perceptually optimum VFR transmission. The parameter error definitions we have used in our implementation are given by the following equations, where the symbols P(n), G(n), and $\underline{g}$(n) are used to denote, respectively, pitch period in number of samples, energy expressed in dB, and LAR vector, all for the nth frame in the block. Primes are used to indicate quantized values for transmitted frames and linearly interpolated values for untransmitted frames.

$$E_p = \sum_{n=1}^{N} [P(n)-P'(n)]^2, \qquad (17)$$

$$E_g = \sum_{n=1}^{N} [G(n)-G'(n)]^2, \qquad (18)$$

$$E_c = \sum_{n=1}^{N} W(n)d(\underline{g}(n),\underline{g}'(n)), \qquad (19)$$

$$d(\underline{g}(n),\underline{g}'(n)) = \sum_{i=1}^{N} [g_i(n)-g_i'(n)]^2 \qquad (20)$$

$$W(n) = G(n)/GMAX, \qquad (21)$$

$$GMAX = \underset{n}{MAX}\ G(n). \qquad (22)$$

From (17) and (18), the pitch and energy errors are computed as the sum over the block of the squared frame differences [P(n)-P'(n)] and [G(n)-G'(n)], respectively. The LAR error is computed as the energy-weighted Euclidean distance as shown in Equations (19) to (22). In addition, one may empirically determine thresholds on the individual parameter errors: When a parameter error computed at any frame in a sequence exceeds its threshold, that sequence is rejected. In our work, we set w_p in (16) to zero, and used the latter thresholding method to reject a frame sequence that yields at any frame a pitch error whose magnitude exceeds a prespecified threshold T_p.

The set of allowable sequences is determined not only by the block size N and the number of transmitted frames M, but also by the following constraints:

- Always transmit the last frame in a block (to reduce the interpolation delay at the receiver).
- Always transmit the first voiced frame in a U-V (unvoiced-voiced) transition.
- Always transmit the last voiced frame and the first unvoiced frame in a V-U transition.

Clearly, the VFR scheme described above introduces time delay in the speech communication link. The time delay introduced at the transmitter is given by the duration of the block, i.e., N frames. Since the last frame in a block is always transmitted, the receiver time delay is (N-M+1) frames. To determine the overall delay between input and output speech, one must add to the foregoing two delay components two additional components required by the underlying fixed frame rate analysis and synthesis: one due to LPC analysis and pitch extraction (2 frames) and the other due to frame-to-frame interpolation required at the synthesis (1 frame). Therefore, the overall delay is (2N-M+4) frames.

Since we assume that spectral deviations are transmitted every frame, we first perform the VFR analysis and then compute the spectral deviations using the quantized parameters (pitch and LARs) for the transmitted frames and the interpolated parameters for the untransmitted frames.

The program organization of the block-encoded VFR scheme that we implemented and incorporated into the HDV coder simulation is as follows. At every frame, one frame of input speech is analyzed at the transmitter and one frame of output speech is synthesized at the receiver. The frame analyzed is the present frame, and the frame synthesized is N frames into the past, where N is the number of frames in the block. After N consecutive frames have been analyzed, the sequence of M frames to be transmitted is determined. At this time, the program performs the following operations: computation of spectral deviations, encoding of the transmitted data, decoding of the same, and interpolation to produce N frames of data. These N frames of data are made available to the receiver "instantly." It is as though there was no receiver delay introduced by the VFR scheme. We did not simulate the true receiver delay of (N-M+1) frames, to limit the program complexity.

We now consider an example to give a breakdown of the bits used per block in a 2400 b/s HDV coder. Let the block size (N) be 7 frames and the number of frames transmitted per block (M) be

5 frames. Since the last frame in a block is always transmitted, we have 15 possible frame sequences and we need a 4-bit block header. For channel error protection, the header may be transmitted in three copies, and this requires 12 bits for the protected header. The total number of bits available per block is given by: 7 (frames) x 20 (ms) x 2.4 (kb/s) = 336 bits per block. The bits used for the various parameters per frame may be allocated as follows:

LARs	:	35
Pitch	:	6
Gain	:	5
Voicing	:	1
Sync	:	1

Total = 48 bits/frame

Since there are 5 transmitted frames per block, the total number of bits used is 12+5x48=252. This would leave 336-252=84 bits per block, to be used for spectral deviations and error protection of parameters. For instance, one could transmit 6 deviations per frame at 2 bits each, for all 7 frames, using 12x7=84 bits. This system would not leave any bits for channel error protection of parameters. However, if we change N and M to be 8 and 5 frames, respectively, and keep all other aspects as given above, it can be seen that the (5,8) system would provide 30 bits per block or 6 bits per frame for error protection.

We determined, through experiments, the values of the three decision parameters of the VFR scheme: the weight w_g of the energy error [see (16)], the weight w_c of the LAR error, and the pitch-error threshold T_p. [Recall that we set w_p to zero in (16).] In these experiments, we used an analysis rate of 50 frames/s and considered several (M,N) pairs. Of the several choices we investigated, the choice w_g=1.0 and w_c=0.1 produced the best speech quality. With this choice, the pitch error at any frame was found to be rather small. In all our experiments reported below, we used T_p=10 samples. In the cases we examined, this threshold was never exceeded. We note that the LAR error E_c was about 50 times larger than the energy error E_g, and that moderate variations in the values of w_g and w_c did not change the output speech quality, indicating the applicability of w_g=1.0 and w_c = 0.1 over a broader range of conditions than we considered in the above empirical study. Therefore, we used this choice in all our subsequent work.

9.2 A Tradeoff Study

We performed a limited tradeoff study to understand the effect of block length (N frames) and transmission frame rate (proportional to M/N) on the output speech quality of the above VFR system. In this study, we used an analysis rate of 50 frames/s and transmitted either no spectral deviations (LPC coder) or 6 deviations (HDV coder) using the spectral peak adaptive method described in Section 7.3. The various systems we simulated are described in Table 7. For these systems, we used 48 bits per frame for transmitting all parameters except the deviations: 35 bits for 12 LARs, 6 bits for pitch, 5 bits for energy, and 1 bit each for voicing and synchronization. The first column in Table 7 indicates the (M,N) value of each VFR system, and the second and the third columns give, respectively, the corresponding block rate in blocks/s and the average transmission frame rate in frames/s. The fourth column gives the value of the total time delay between the input speech and the output speech in milliseconds. The fifth column gives the number of bits required for the block header. We have assumed that three copies of the block header are transmitted for protection against bit errors. The next three columns give the total transmission bit rates for three cases: (a) LPC vocoder (no harmonic deviations), (b) HDV coder with deviations transmitted every frame, and (c) HDV coder with deviations transmitted at VFR. The case (c) is treated below in Section 9.5. A specific goal of this tradeoff study was to look for systems that yield the best output speech quality at bit rates around 2400 b/s and 2000 b/s. The lower bit-rate systems would be used with error protection in applications involving channel errors.

We compared the speech quality of the VFR coders shown in Table 7 through informal listening tests. The results of these

System ID (M,N)	Block Rate (Blocks/s)	Frame Rate (Frames/s)	Time Delay (ms)	Header Size (Bits)	Transmission Bit-Rate (b/s)		
					(a) No Devns. (LPC)	(b) Devns. every frame	(c) Devns. at VFR
(4,6)	8.33	33.3	240	4	1699	2299	2099
(5,7)	7.14	35.7	260	4	1800	2400	2229
(5,8)	6.25	31.3	300	6	1613	2213	1988
(6,8)	6.25	37.5	280	5	1894	2494	2344
(6,9)	5.55	33.3	320	6	1700	2300	2100
(7,9)	5.55	38.8	300	5	1950	2550	2417
(6,10)	5.00	30.0	360	7	1545	2145	1905
(8,12)	4.17	33.3	400	9	1712	2312	2112

Table 7. Description of VFR systems included in our tradeoff study.

tests are summarized as follows. All VFR-HDV coders tested produced better speech quality than did their LPC counterparts. In general, better speech quality was obtained with a lower block rate (larger N) and with a higher frame rate (larger M/N). But, a low block rate causes a long time delay, and a high frame rate requires a high bit rate. At a given frame rate, a lower block rate system produces better speech quality, since the associated longer block size provides more freedom in the selection of the frames to be transmitted, as compared to a higher block rate system. The systems (4,6), (5,7), and (5,8) all had perceivable quality distortions that were not produced by fixed frame rate systems. Of these three systems, the (5,8) system produced better overall speech quality than the other two, even though it uses a lower frame rate. All other VFR systems, shown in the last 5 rows of Table 7, produced speech without any of the distortions mentioned above. From the results obtained from systems (5,8) and (6,8), we note that the increased frame rate of the latter system compensates for the moderately high block rate of 6.25 blocks/s. The importance of a low block rate is further strengthened by our preference of the systems (6,9), (6,10), and (8,12) over the system (5,7), although the latter system transmits at a higher frame rate and bit rate than the other three systems. The systems (6,8) and (6,9) produced speech with about the same speech quality, indicating that the benefits of high frame rate can be traded for the benefits of low block rate. Compared to the (6,9) system, the (6,8) system has the advantage of a lower time delay and the disadvantage of a higher bit rate. For the same block rate, the (7,9) system produced better speech quality than did the (6,9) system, but the former requires a higher bit rate. Next, we compared the (6,9) system first with the (6,10) system and then with the (8,12) system. System (6,10) produced speech that was slightly inferior in quality to speech from system (6,9), indicating the importance of a sufficiently high frame rate. Systems (6,9) and (8,12) yielded about the same speech quality, at the same frame rate and about the same bit rate; this result indicates that there is not much to be gained in decreasing the block rate below about 5 blocks/s. Of course, system (6,9) is preferred over system (8,12), since the latter requires an unacceptably large time delay.

From the results presented above, we recommend a block size of 150 ms or longer and an average transmission frame rate of 33 frames/s or higher. The three systems that satisfy these recommendations are (6,8), (6,9), and (7,9). (The (8,12) system is not recommended because of the comments made in the last paragraph in reference to its comparison against the (6,9) system.) Compared to the fixed frame rate (50 frames/s) LPC vocoder, all three systems with no deviations produced the same or better speech quality; better quality was obtained primarily during slowly varying speech, for which the FFR system produced a "wobble" quality and each VFR system produced a smooth and a more natural speech [28]. All three VFR-HDV coders produced noticeable improvements in speech quality over the FFR-LPC coder.

Since the (4,6) system requires a significantly shorter time delay than the above recommended systems do, we investigated two methods of reducing or eliminating the distortions produced by this system: (1) remove the constraints that force the transmission of frames in voicing transitions (Section 9.3) and (2) use a 100 frames/s analysis frame rate to provide a better time resolution for the selection of frames to be transmitted (Section 9.4). We hasten to point out that both methods did not produce the improvement we sought, but they are treated in the next two sections in view of their relevance to the recommended systems.

9.3 Constraints Due to Voicing Transitions

In the block-encoded VFR scheme described above, we use the reasonable constraints that force the transmission of the voiced frame in an unvoiced-voiced (UV) transition and of both frames in a VU transition. To examine if these constraints cause a suboptimal transmission and hence inferior speech quality, we investigated the case in which these constraints are removed.

The unconstrained system allows sequences like UuvvV or VvuuU, where uppercase letters indicate transmission frames and lowercase letters indicate frames to be interpolated. In such situations the question arises as to how to regenerate the parameters of the untransmitted frames in the transition region. In this investigation, we assumed that the voicing status (1 bit) of each frame is known at the receiver. We considered several regeneration schemes: linear interpolation between voiced and unvoiced frames, a stepwise constant model where the transmitted frame is copied forward or backward onto the region of frames that have the same voicing status, and a combination of these two models, in which the stepwise constant model is used during the unvoiced part of the transition and linear interpolation is used during the voiced part. Of all cases we investigated, the stepwise constant model was found to yield the smallest parameter error. However, in listening tests, we preferred the system with linear interpolation, since it did not have the degradations perceived in the other cases. In a detailed examination of the transmission decisions made by the unconstrained system (using any of the regeneration methods), we found that the system, in general, transmitted the first voiced frame in a UV transition and both frames in a VU transition. This finding confirms that our original constraints due to voicing transitions are in fact good and do not cause any ill effects. Since such constraints alleviate the computational load in VFR by eliminating some of the sequences to be searched, we have decided to retain them in our VFR system. Also, with these constraints in effect, the voicing bit is required for the transmitted frames only.

9.4 100 Frames/s Analysis Rate

In another attempt to improve the speech quality in VFR systems, we investigated the use of 100 frames/s analysis rate instead of the 50 frames/s rate used in our work thus far. The motivation for this study is that a 100 frames/s rate affords an improved time resolution, which may alleviate some of the distortions present in some VFR systems. The effect of using 10 ms frames, produced by the 100 frames/s analysis, is to double the block size N. We denote such systems by (M/N); e.g., system (4,6) with 20 ms frames can now be implemented as system (4/12) with 10 ms frames. Doubling the block size results in a large increase in the number of possible sequences of M frames that have to be searched to determine the best one to transmit. For example, system (4/12) has 165 possible sequences in each block, and system (6/18) allows for 6188 sequences! We narrowed our investigation to system (4/12), since it produces a significantly shorter time delay than do systems like (5/16) and (6/18), and requires a substantially lower computational expense of searching over possible frame sequences. Informal listening tests, however, showed that the speech from system (4/12) was no better than from system (4,6). We note also that the implementation of the VFR system using 10 ms frames requires about twice the memory or storage space and a 5-fold increase in computations as compared to the 20-ms VFR system. This last consideration is quite important since the final algorithm must be made to run on the sponsor's PDP-11. Therefore, we decided not to consider 100 frames/s analysis in our subsequent work.

9.5 VFR Transmission of Spectral Deviations

In this method, we transmit the deviations at a variable frame rate, together with the coder parameters. In other words, the deviations are transmitted only at those M frames for which the coder parameters are also transmitted in (M,N) VFR systems. The deviations at the untransmitted (N-M) frames are computed at the receiver by a process of spectral interpolation, which we discussed in Section 6.4. We illustrate this method by means of the following example. Assume that frames 1 and 4 are transmitted, with frames 2 and 3 to be interpolated. Let L(i) denote a vector of LARs and D(i) a set of spectral deviations, at frame i. Thus, L(1), D(1), L(4), and D(4) are known at the receiver. From L(i) and D(i), the receiver computes a "total" spectrum S(i), which is the sum (in dB) of the LPC spectrum H(i) (which is computed from L(i)) and the deviations D(i). Thus, the receiver computes S(1) and S(4) for the transmitted frames, and finds the interpolated total spectra S'(2) and S'(3) by linear interpolation between S(1) and S(4). Also, the receiver computes the linearly interpolated LAR vectors L'(2) and L'(3) and their corresponding spectra H'(2) and H'(3). The differences D'(2)=S'(2)-H'(2) and D'(3)=S'(3)-H'(3) are computed at the harmonics of the respective fundamental frequency (or at multiples of 100 Hz, for unvoiced frames) to produce the required spectral deviations for frames 2 and 3. In computing the spectrum of the LPC filter, we used a high-order (1024-point) FFT.

For the different VFR-HDV coders shown in Table 7, VFR transmission of the deviations reduces the bit-rate by 133-240 b/s, as seen by comparing the two right-most columns in that table. For the three systems (6,8), (6,9), and (7,9), we compared the two cases: transmission of the deviations every frame and VFR transmission. We found through informal listening tests that there was no perceivable quality degradation in going from a system where the deviations were transmitted at all frames to a system where they were transmitted at VFR. This last result is important because, without any speech quality degradation, we achieved a reduction in bit rate by 133-200 b/s. With this reduction in bit rate, we can consider the (6,8) and (7,9) systems for use over noise-free 2.4 kb/s channels and the (6,9) system for use over noisy 2.4 kb/s channels.

With the spectral deviations transmitted at VFR, one has the option of computing the deviations (1) every frame, before making the VFR transmission decisions, or (2) only for the transmitted frames, after the VFR analysis. Case (1) involves more computation but requires the storage of only a small number (e.g., 6) of selected deviations for every frame in the block. On the other hand, case (2) requires the storage of the speech (or the residual) samples themselves (200 samples per frame) for all the frames in the block. The requirement of excessive storage in case (2) will make it difficult to fit the simulation of the final HDV coder within the limited address space available on the sponsor's PDP-11. In addition, with the option (1) above, the operations of analysis and transmission are clearly separated, which allows them to be coded into separate modules. From these considerations, we modified our simulation program to implement the option (1) above.

1Ø. OPTIMIZATION OF HDV CODERS FOR ERROR-FREE CHANNELS

Although the final goal of this work has been to develop a robust HDV coder for use over noisy channels, initially we conducted the speech-quality optimization study for error-free channels to investigate the speech quality that the HDV coders are capable of producing at 2.4 kb/s, without the burden of the error protection bits. Also, we felt that parameter tradeoff relations obtained in this study could be used in narrowing the range of parameter values to investigate in the subsequent optimization study for noisy channels. The results of the study reported in this chapter and the recommendations given at the end of Section 1Ø.6 should be clearly useful in the design of 2.4 kb/s systems for speech communication over error-free channels.

In the preceding chapters, we have made a number of recommendations concerning the various aspects of HDV coders including extraction, selection, and coding of spectral deviations, LPC order, quantization of log area ratios, VFR transmission of the coder parameters, and synthesis. These recommendations are summarized in Section 1Ø.1. In Sections 1Ø.2 to 1Ø.5, we consider specific design parameters of the HDV coder and report the speech quality effect of varying them individually around their normal values used in our investigations prior to this optimization study. Section 1Ø.5 also contains the results on the performance of HDV coders under speech inputs that have been bandlimited to a frequency significantly below the normal value. In Section 10.6, we present several 2.4 kb/s HDV coders that we selected using the results reported in Sections 10.1 to 10.5, and compare their speech quality. From the results of these comparisons, we obtain the optimized HDV coder for error-free channels.

10.1 Overall System Specification

The HDV coder we recommend has the following overall specifications: block-encoded VFR transmission of all the transmitted data, with a block size of 150 ms or longer and an average frame rate of 33 frames/s or higher; LPC order p=12 for input speech with a bandwidth of 5 kHz (see Section 10.5 for the treatment of 4-kHz bandwidth); quantization of the LARs using the optimal scalar method; extraction of the spectral deviations using the smoothing-sampling method, which employs cepstral smoothing and F_0 (100 Hz for unvoiced speech) sampling; transmission of a subset of the deviations at 2 bits each, using the spectral peak adaptive method described in Section 7.3; and pitch-synchronous zero-phase reconstruction of the excitation signal (random-phase reconstruction once every 10 ms, for unvoiced speech) using the interpolated spectral deviations as described in Section 6.4.

10.2 Number of Bits for LAR Quantization

From the results reported in Section 8.4, the reasonable choices for the number of bits for LAR quantization are: 35, 38, and 41 bits per frame. We simulated the VFR-HDV coders (6,8), (6,9), and (7,9) given in Table 7, using each of these three values for LAR quantization. Assuming VFR transmission of 6 deviations, the bit rates for the three cases are 2344, 2456, and 2569 b/s for the (6,8) system; 2100, 2200, and 2300 b/s for the (6,9) system; 2417, 2533, and 2650 b/s for the (7,9) system. In each of the three VFR systems, we found that there was a slight but perceivable speech quality improvement in going from 35 to 38 bits per frame for LAR quantization and a negligible change in quality in going from 38 to 41 bits per frame.

10.3 Analysis Rate

The normal value we used for the analysis rate is 50 frames/s. In Section 9.4, we reported that using a 100 frames/s analysis rate did not improve the quality of a VFR coder, but it increases substantially both the computational complexity and the storage required, as compared to the 50 frames/s analysis. We also investigated another analysis rate: every 22.5 ms or 44.44 frames/s. The 22.5 ms frame size offers some compatibility with the present Government standard 2.4 kb/s vocoder LPC-10, and leads to a lower transmission frame rate. We conducted several experiments in which we compared 22.5 ms systems with 20 ms systems. The results are summarized below.

a. Considering fixed frame rate LPC vocoders and using 35 bits for LAR quantization, we found that the 22.5 ms system at 2133 b/s produced speech that was significantly degraded compared to the 20 ms system at 2400 b/s.

b. The speech quality of the above 22.5 ms system improved greatly when we transmitted 3 spectral deviations at 2 bits each. The resulting 22.5 ms HDV coder has a bit rate of 2400 b/s, and it produced slightly better speech quality than did the 20 ms LPC coder using the same bit rate.

c. We implemented a 22.5 ms (7,9) VFR-HDV coder, using 35 bits for LARs and 6 deviations. This coder requires an average transmission frame rate of 34.6 frames/s and a bit-rate of 2148 b/s. A comparable 20 ms coder is the (6,9) VFR-HDV coder, which requires 33.3 frames/s and 2100 b/s. In comparing the speech quality of the two coders, we preferred the 20 ms coder over the 22.5 ms coder.

d. In another experiment, we added two fixed (untransmitted) spectral deviations (see Section 7.3) to the 20 ms, 2400 b/s FFR-LPC coder (see (a) above). This coder provided speech quality that was, in fact, superior to the quality of the 22.5 ms, 2400 b/s FFR-HDV coder with 3 transmitted deviations (see (b) above).

From the results presented above, we decided to continue to use the 50 frames/s analysis rate.

10.4 Number of Transmitted Deviations

Thus far, we have considered VFR-HDV coders that transmit 6 deviations adaptively, of which the first two are located at the fundamental frequency and the first harmonic, and the remaining four are located, in pairs, around the first and second peaks of the LPC spectrum. In one experiment, we compared systems that had 0,2,3,4,6, and 8 deviations, respectively. For this comparison, we used the (6,9) VFR system with 35-bit LAR quantization. Through informal listening tests, we found that the largest increment in speech quality occurred in going from 0 to 2 deviations. Also, there was another noticeable increment in speech quality in going up to 3 or 4 deviations. However, we observed little or no additional improvement in quality in using more than 4 deviations. In another experiment, aimed at simplifying the computations, we simulated systems transmitting the first n deviations, with n=3 and n=4. The comparison of systems with first n deviations to systems with n adaptive deviations showed that the adaptive cases yielded a small improvement in speech quality. If the computations required by the adaptive deviations should be a problem, one can use the first n deviations with a small sacrifice on speech quality. In all our subsequent optimization studies reported below and in the next chapter, we have used the adaptive deviations only.

1Ø.5 Input Speech Bandwidth

In all our experiments on HDV coders reported thus far, we used a 5-kHz bandwidth and a 1Ø-kHz sampling rate for the input speech. As part of our optimization study, we investigated the HDV coder using a 4-kHz bandwidth and an 8-kHz sampling rate. The use of 8-kHz sampling rate instead of 1Ø kHz implies that we need about 2 fewer (1Ø instead of 12) coefficients for the LPC spectral model. This saving may be used either to increase the quantization accuracy of LARs or to increase the number of bits allocated for error protection. That the present government standard 2.4 kb/s system LPC-1Ø uses 8-kHz sampling rate provides another reason for our work on 8-kHz HDV coders -- one that would permit a simple tandem link between the HDV and LPC-1Ø coders. The 1Ø-kHz system, on the other hand, has the potential of yielding better speech quality and intelligibility because of the information the system transmits in the frequency range from 4 to 5 kHz.

In Section 1Ø.5.1 we describe our work on the modification and testing of the HDV coder to process speech sampled at 8 kHz. The modification to the coder involved the design of optimal LAR quantizers for the 8-kHz speech signal. Our goal in the testing was to compare the quality of HDV coders based on 8-kHz speech and on 1Ø-kHz speech. In Section 1Ø.5.2, we treat a related issue: the performance of 10-kHz HDV coders with 4-kHz bandlimited input speech.

10.5.1 Design and Testing of 8-kHz HDV Coders

We computed the statistics required for eigenvector rotation and MMSE quantization of LARs, using a training database of 12 sentences recorded from 3 males and 3 females, lowpass filtered at about 3.8 kHz, and sampled at 8 kHz. This database was used in a previous government-sponsored contract at BBN [26, 27]. We then designed optimal scalar quantizers for LARs for the following cases:

- 10-pole case: 33, 35, and 38 bits/frame
- 12-pole case: 35 and 38 bits/frame

As with 10-kHz HDV coders, we used as test database the set of 6 phoneme-specific utterances. Since these utterances were originally sampled at 10 kHz, we resampled them digitally at 8 kHz. This resampling involved three steps: upsampling the 10-kHz speech to a rate of 40 kHz, lowpass filtering at 4 kHz, and downsampling to the 8-kHz rate. For the lowpass filtering process, we designed a 256-point FIR filter using a Hanning window on an ideal lowpass filter.

We conducted several experiments comparing 8-kHz HDV coders against each other and against 10-kHz coders. The results are summarized below.

8-kHz Coders:

a. Considering 10-pole fixed-frame-rate coders, we obtained about the same speech quality using 33, 35, or 38 bits/frame for LARs.

b. At 35 bits/frame, we preferred the 10-pole system over the 12-pole system.

c. The 12-pole 38 bits/frame system produced only a small improvement over the 10-pole 33 bits/frame system.

8-kHz Coders vs. 10-kHz Coders:

d. At 35 bits/frame, we compared 2 systems: 1) 10-pole, 8-kHz system and 2) 12-pole, 10-kHz system. System 2 produced speech that was more "crisp" and had perceivable, though noisy, high end. System 1, on the other hand, produced less low-frequency roughness.

e. The 10-kHz system that transmits 12 LARs at 38 bits/frame produced better speech quality than any of the five 8-kHz systems we tested.

f. We simulated the (6,9) VFR-HDV coder given in Table 7, using 8-kHz sampled speech and 33-bit quantization of 10 LARs, and compared it with the corresponding 10-kHz system (35 bits for 12 LARs). The speech quality of the 10-kHz system was found to be superior to that of the 8-kHz system.

10.5.2 Bandlimited Input Speech

There is concern that the HDV coder must be able to process speech bandlimited to 4 kHz, as this is encountered in tandem operation with LPC-1Ø (or other waveform coders). To evaluate the performance of our 1Ø-kHz HDV system under the constraint of a bandlimited input, we performed several experiments. We digitally lowpass filtered our 6-utterance database to 4 kHz, keeping the sampling rate at 1Ø kHz. The new, bandlimited database was processed by the (6,9) system (see Table 7) under the following cases: a) 14-pole LPC analysis without quantization of any of the parameters, b) 12-pole analysis without quantization, and c) 12-pole analysis with quantization of all parameters. The outputs from these cases were compared to the output of the corresponding 8-kHz VFR system (see (f) above). We ran the 8-kHz system twice, once without parameter quantization and once with parameter quantization. First, we considered the 1Ø-kHz systems and compared the full-band case (5 kHz) to the bandlimited case (4 kHz). We observed only a slight degradation in the quality of the output speech due to the bandlimited input speech, in both the quantized and unquantized cases, and whether the output D/A lowpass filter was set to a cutoff of 4 kHz or 5 kHz. The 14-pole and 12-pole systems produced similar speech quality. The results of this experiment suggest that the HDV coder is robust against the condition of bandlimited input. Second, considering an input bandwidth of 4 kHz, we compared the 1Ø-kHz systems to the 8-kHz systems. We found the outputs of the two systems to be similar in quality, in both unquantized and quantized cases. When the input speech was not bandlimited, we consistently preferred the 10-kHz system over the 8-kHz system.

10.6 Comparison of Selected 2.4 kb/s HDV Coders

The results presented in the preceding sections show that the optimized 2.4 kb/s coder must have the following: 5-kHz input speech bandwidth and 10-kHz sampling rate, 12-pole LPC analysis, and 50 frames/s analysis rate. To determine the tradeoff between the remaining parameters, we selected five 2.4 kb/s VFR-HDV coders. We have assumed that the block header is transmitted in three copies. The five coders, which are described below, are denoted using a 3-digit number MNn, where M, N, and n are, respectively, the number of transmitted frames per block, the block length in frames, and the number of transmitted deviations.

685 : (6,8) system, at 38 bits for LARs and 5 deviations

684 : (6,8) system, at 41 bits for LARs and 4 deviations

698 : (6,9) system, at 41 bits for LARs and 8 deviations

696 : (6,9) system, at 38 bits for LARs and 6 deviations at FFR

795 : (7,9) system, at 38 bits for LARs and 5 deviations

These coders involve three VFR systems: (6,8), (6,9), and

(7,9). For the first two VFR systems, we consider a tradeoff between quantization accuracy for the LARs and the number of transmitted deviations. In going from one VFR system to another, we examine the effect of changing the average frame rate and the block rate. Upon listening to the speech from the various systems, we found that they all produced very similar speech quality. Systems 684 and 685 were very close. We preferred system 698 over 696. We chose the systems 685, 698, and 795 for further comparisons, and ranked them in the order of decreasing quality as follows: 795, 685, and 698. The specification of the best system 795 is given in Table 8, where we have provided the breakdown of bits. (Notice that instead of using 1 sync bit every frame, System 795 uses 4 bits every block or 1 bit every 1Ø8 bits.)

We also included in our comparisons two fixed frame rate, 2.4 kb/s systems, both using 35 bits per frame for LARs: 1) the LPC vocoder and 2) the "fixed" HDV coder that uses only 2 deviations that are fixed for all frames (i.e., not transmitted). We found that the five VFR-HDV coders produced better speech quality than did the fixed HDV coder. The fixed HDV coder, in turn, produced speech quality superior to the LPC vocoder.

From the view of achieving the best speech quality, we recommend the VFR-HDV coder 795. This optimal system, however,

Block length, N		= 9 frames
Number of transmitted frames per block, M		= 7 frames
Block rate		= 5.55 blocks/s
Frame rate		= 38.8 frames/s
Time delay		= 300 ms
Available bits per block		= 432 bits
One frame:		
12 LARs	= 38 bits	
Pitch	= 6 bits	
Gain	= 5 bits	
5 Deviations	= 10 bits	
Total bits/frame	= 59 bits	
Total parameter bits/block		= 413 bits
Header (3 copies)		= 15 bits
Sync bits per block		= 4 bits
Total bits used per block		= 432 bits

Table 8. Bit allocation for the optimized 2.4 kb/s HDV coder for noise-free transmission (system 795).

introduces a time delay of 300 ms between the input and output speech. For applications where the time delay must be kept at an absolute minimum and where system complexity must be reduced, we recommend the "fixed" HDV coder.

11. OPTIMIZATION OF HDV CODERS FOR NOISY CHANNELS

One of the requirements of this project has been to design the 2.4 kb/s HDV coder so that it transmits good quality speech over noise-free channels and exhibits a graceful degradation in the presence of random channel bit-errors of 1%. In our study to meet this objective, we used the Hamming (7,4) code, which protects 4 data bits by adding 3 parity bits; this code detects and corrects all single bit-errors in the resulting 7-bit codeword. The bits of the block header and the voicing bit are protected by simple redundancy code, i.e., by repeating each bit serveral (odd number of) times. At the receiver, a simple majority rule is used for decoding. Clearly, using a (2n+1)-bit redundancy code for a bit protects it against up to n single bit-errors. We used the results that we obtained previously in a government-sponsored project at BBN, involving a robust 2.4 kb/s LPC vocoder [26, 27]. After some investigation of HDV coders in channel error, we selected and simulated several candidate 2.4 kb/s HDV coders, compared their speech quality in both 0% and 1% error, and decided on the final optimized coder.

11.1 Channel-Error Simulation

In our simulation, we used the binary symmetric channel in which independent, identically distributed random errors are introduced into the transmitted bitstream. A bit error simply changes the bit from 0 to 1 or from 1 to 0. The software developed is quite versatile in that we may set the percentage of error as well as the degree of protection for each parameter independently. With this approach, it is possible to determine which parameters, when subjected to channel error, are the major deterrent to good overall coder performance. The allocation of protection for each parameter may also be evaluated in a similar manner. For careful study and diagnostic purposes, we implemented a facility to print out the transmitted code and the received and error-corrected code for each of the quantities, header, voicing bit, pitch, energy, LARs, and deviations, whenever the two are not the same because of one or more bit-errors. Because of the random nature of the bit-errors, two coders with different bitstreams will have their parameters exposed to, in general, different bit-error patterns even if the random number generator is intitialized at the same value at the start of each coder operation. Thus, comparison of two coders cannot be reliably done on a sentence-by-sentence basis. Speech from each system should be evaluated separately to get an overall impression, which can then be compared against the overall impression of another competing system. To facilitate this comparison process, we processed ten sentences (rather than six) for all of the important comparisons we made. (The additional 4 sentences are the all-voiced sentences used in our work on the pitch-synchronous HDV coder, reported in Chapter 5.)

Finally, we equalized the LAR bit allocations for voiced and unvoiced cases, for two reasons: 1) This prevents possibly large LAR decoding errors in case of a decoding error in voicing; and 2) with the same bit allocation, channel-error simulation can be done without actually creating a bitstream of the transmitted frames. This reduces greatly the complexity of the program. We achieved the equal bit allocation by simply forcing the allocation for unvoiced speech to be the same as for voiced speech. This step involved changing the bits assigned to two coefficients (located after the 4th LAR), one increased by 1 bit and the other decreased by 1 bit; we then recomputed the quantization tables for the two coefficients. We did not observe any perceivable speech quality change due to the foregoing equalization. As it happens, some error-protected coders can use only 37-bit (instead of 38-bit) LAR quantization (see below). So, we designed LAR quantizers using a total of 37 bits. The bit allocation we used is (5,4,4,3,3,3,3,3,3,3,2,1) for the 37-bit case and (5,4,4,3,3,3,3,3,2,2,2,1) for the 35-bit case.

11.2 Error Protection of Coder Parameters

From our previous work on the performance of 2.4 kb/s LPC coders in channel error, we know that triple redundancy protection of the block header and use of 3 Hamming (7,4) codewords to protect 4 most significant bits (MSB's) each of pitch and energy and 2 MSB's each of the first two LARs produce adequately robust performance in channel error [26, 27]. Decoding errors in voicing, due to channel bit-errors, are detrimental to speech quality in any coder. They may be particularly bad in our HDV coder since we are using different LAR quantization for voiced and unvoiced frames. Again, our previous work has shown that sending 5 copies of the voicing bit provides a reliable performance.

As a first step in studying the channel-error performance of HDV coders, we performed an experiment to evaluate the speech-quality effects of channel bit-errors on the harmonic deviation data. In this experiment, we simulated the (6,9) VFR system (see Table 7) under 4 conditions: a) 0% error on all parameters; b) 1% error on all parameters; c) 0% error on all parameters except the 6 deviations, which were exposed to 1% error; and d) 0% error on the deviations and 1% error on the rest. As a reference, we simulated the FFR-LPC coder at 0% error and 1% error. From comparisons of these six cases, we obtained the following results:

- Bit-errors on the deviations do not have any serious impact on the speech quality (certainly no more serious than caused by errors on LARs)
- The HDV coder produces noticeable improvement in speech quality over the LPC coder even in 1% error
- Error-protection of deviations is not mandatory for a robust performance.

11.3 Experimental Comparison of Selected 2.4 kb/s HDV Coders

From the results just presented and those reported in Chapter 10 for error-free transmission, we selected several 2.4 kb/s (including error protection) coders for simulation and detailed comparative evaluation. These systems are described in Table 9. The first four systems are VFR-HDV coders. Systems S1 to S3 are (6,9) systems, provide 3 Hamming codewords for error protection, and represent a tradeoff (in the amount of only 2 bits per frame) between LAR quantization, number of deviations, and header transmission. System S4 is a (6,8) system, transmits only 2 deviations, and provides 2 Hamming codewords for protecting 4 MSB's each of pitch and energy. Systems S5 and S6 are both FFR systems, and they have a data rate of 2.4 kb/s only if we do not count the 2 repetitions of the voicing bit (100 b/s). The 3-bit redundancy protection of voicing protects against single bit-errors, and we used this in an attempt to approximate the voicing protection (against single bit-errors) provided by the 7-bit pitch and voicing code that LPC-10 uses [12]. (Notice that the 3-bit code we use gives better protection than the LPC-10's 7-bit code does, because, while both codes provide protection against single bit-errors only, our code is shorter.)

We performed a detailed comparison of the six coders in 0% error and in 1% error. In 0% error, S4 produced slightly better speech quality than did systems S1 to S3. In 1% error, however, S4 produced more degradations than did S1 to S3. In 0% error, S2 produced a rather small speech quality improvement over S1 and S3. In 1% error, systems S1 to S3 produced equivalent results. It should be pointed out that in 1% error, none of the systems had encountered any decoding errors in header or in voicing. In channel error, all 4 VFR systems performed substantially better than S5 and S6 did. Part of this performance difference is attributed to the partial protection of parameters that the VFR systems provide. Also, we believe that a significant part of this difference is due to the inherent smoothing that the VFR systems perform in the process of regenerating the untransmitted data through interpolation. In channel error, systems S5 and S6 produced similar speech quality.

To check the speech-quality effect of header decoding errors, we simulated S2 and S3 in 3% error. S2 encountered 4

System	VFR Pair	LAR Bits	No. of Devns.	Header Copies	Voicing Bits	Hamming Codewords
S1	(6,9)	35	4	3	5	3
S2	(6,9)	37	3	3	5	3
S3	(6,9)	35	3	5	5	3
S4	(6,8)	35	2	3	5	2
S5	FFR	35	-	-	3	-
S6	FFR	35	(2 fixed)	-	3	-

Table 9. Description of systems tested in channel error.

header decoding errors, over the 10 sentences, while S3 had none. As we anticipated, the header errors caused relatively small distortions. Upon careful comparisons in 3% error, we felt that speech from S3 was just marginally better than from S2. From an overall assessment, however, we recommend system S2 as the final, optimized system. The bit allocation within a block used by S2 is given in Table 10. From this table, it can be seen that the coder S2 devotes 200 b/s for the transmission of spectral deviations and 500 b/s for error protection.

We then compared S2 in 1% error against S2 in 0% error. Speech for the 1% error case contained some audible but mostly low-level distortions such as pops and clicks. By and large, the difference in the speech quality and intelligibility between the two cases was judged to be minor. Thus, system S2 is quite robust against channel error. To see the effect of error protection, we simulated S2 with the error correction at the receiver suppressed for voicing, pitch, energy, and LARs. (Header error-correction was not suppressed.) We found that the uncorrected case produced noticeable quality degradations compared to the error-corrected case. Also, we compared S2 in 0% error and the best noise-free system 795 described in the last section. The speech-quality difference between the two systems was judged to be quite small.

Block length, N		= 9 frames
Transmitted frames per block,M		= 6 frames
Block rate		= 5.5 blocks/s
Frame rate		= 33.3 frames/s
Time delay		= 320 ms
Available bits per block		= 432 bits
One frame:		
12 LARs	= 37 bits	
Pitch	= 6 bits	
Energy	= 5 bits	
3 Deviations	= 6 bits	
Voicing (5 copies)	= 5 bits	
3 Hamming (7,4) codewords	= 9 bits	
Sync bit	= 1 bit	
Total bits/frame	= 69 bits	
Total bits for 6 frames		= 414 bits
Header (3 copies)		= 18 bits
Total bits used per block		= 432 bits

Table 10. Bit allocation for the optimized 2.4 kb/s HDV coder for noisy channels (system S2).

As compared to the LPC coder S5, the recommended HDV coder S2 produces noticeab'e speech quality improvements in 0% error and substantial improvements in 1% error. Finally, we compared the HDV coder S2 against the government standard 2.4 kb/s coder LPC-10. For this comparison, we used a real-time implementation of LPC-10, which was developed, as part of another DCA-sponsored project at BBN, on the CSPI MAP-300 array processor. We found that the HDV coder produces significantly better speech quality than does LPC-10, in both error-free and 1% error cases.

We point out two issues that warrant further investigation. First issue is the use of selective parameter smoothing at the receiver [29, 30]. Large changes in the decoded parameter values between two successive transmissions may be detected as being caused by channel bit-error, and parameter values in such instances are replaced with values from an adjacent transmitted frame or with interpolated values. Such a receiver smoothing scheme is used in LPC-10 [12]. We believe that occasional distortions that the optimized coder produces in 1% error will be reduced or eliminated with the use of selective receiver smoothing.

Second, the optimized coder sends, during unvoiced speech, a 6-bit pitch code containing zeros, 4 MSB's of which are protected by a Hamming (7,4) code that requires 3 extra bits. Since the voicing bit is transmitted reliably, these 9 bits we use for unvoiced frames are merely wasted. We employed this strategy so that error protection can be done in the same manner for voiced and unvoiced frames -- a factor that has kept the program complexity down. Clearly, the 9 bits can be used to protect 12 parameter bits for unvoiced frames, using 3 Hamming (7,4) codewords. This should also improve the robustness of the final coder design.

12. ACOUSTIC BACKGROUND NOISE

We tested the performance of the optimized 2.4 kb/s HDV coder for input speech corrupted by acoustic background noise (about 6Ø dB SPL) typical in an office environment. (The noise is due to such things as typewriter and coughing or low-level speech from other speakers.) For this test, we used the six sentences from the office-noise database described in Section 3.3.3. When we processed these sentences through the optimized 2.4 kb/s HDV coder, we observed a large number of pitch and voicing errors. A close examination of the AMDF-DYPTRACK program showed that the initial values we used for several adaptive parameters were not appropriate for the sentences from the office-noise database. These values were computed for high-quality speech (see Section 3.2). We resolved the pitch problem as follows. We processed the sentences from males and females in separate runs of the program. In each run, the first sentence was processed twice, with the second processed output written onto a file, which was later used in listening tests. The adaptive parameters seemed to reach reasonable values at the end of the first processing, and from then on, they were allowed to vary continuously from one sentence to the next; in other words, the parameters were not reinitialized at the start of each new sentence. Of course, a real-time implementation of the coder will not have the above problem, since those parameters will adapt to proper values in a few seconds from the time the speaker starts talking.

Using the same procedure, we also processed the six sentences through the 2.4 kb/s LPC vocoder. We then compared the output speech from the LPC and HDV coders. We found that the HDV coder produced noticeable speech-quality improvements over the LPC vocoder. We felt that the amount of improvement was less in the office-noise case than in the high-quality database that we have been using thus far. The reason for this reduction is that the HDV coder has been optimized for the high-quality speech input. Also, the statistics used for the quantization of LARs, energy, and deviations were collected over high-quality speech databases. Since the LPC spectrum for the noise-corrupted speech has, in general, spurious peaks (that do not correspond to formants), the 3rd spectral deviation transmitted by the HDV coder may not correspond to a formant. Had we optimized the HDV coder for the office-noise database, we might have decided to transmit more than 3 deviations.

13. TANDEMING WITH A CVSD CODER

To evaluate the performance of the CVSD-HDV tandem link, we tested the optimized HDV coder using as input speech the six sentences from the CVSD database described in Section 3.3.4. Using the procedure explained in the previous section, we processed these sentences using the optimized HDV coder. As expected, the quality of the output speech was found to be inferior to that of the input CVSD speech. However, informally comparing the intelligibility in the two cases, we felt that there was no major reduction in the intelligibility at the output of the tandem. We also processed the six CVSD sentences through the 2.4 kb/s LPC vocoder. We preferred the HDV coder output over the LPC coder output. The remarks made at the end of the previous chapter can be repeated here as well.

We believe that the speech intelligibility of the HDV-CVSD link will be only slightly lower than the intelligibility of the (single-link) HDV coder. We did not, however, test this tendem link.

14. DESCRIPTION OF THE OPTIMIZED, 2.4 KB/S HDV CODER

Figures 10 and 11 show a block diagram of the optimized coder. Table 10 in Chapter 11 provides information regarding quantization, error protection, and VFR transmission of the parameter data of the HDV coder. At the transmitter, the analog input speech is lowpass filtered at 5 kHz and sampled at 10 kHz. Referring to Fig. 10, the sampled speech s(t) is divided into non-overlapping frames of 200 samples (20 ms duration). Each frame of speech is subjected to three types of analyses: linear prediction, pitch and voicing, and spectral deviations. Considering LPC analysis, the dc value over the frame is removed from the input speech samples. The energy of the dc-removed samples is computed as their mean-squared value, expressed in decibels, coded, and decoded. The extracted energy in dB, its coded value (integer level), and its decoded (or quantized) value are stored in three separate buffers, which are used in variable frame rate transmission. LPC analysis of the dc-removed speech signal consists of Hamming windowing, computing the autocorrelation function for lags 0-12, using the autocorrelation method to obtain 12 reflection coefficients, and computing log area ratios from the reflection coefficients. The LARs are coded by subtracting their (pre-stored) mean, transforming the zero-mean coefficients using their (pre-stored) eigenvector matrix,

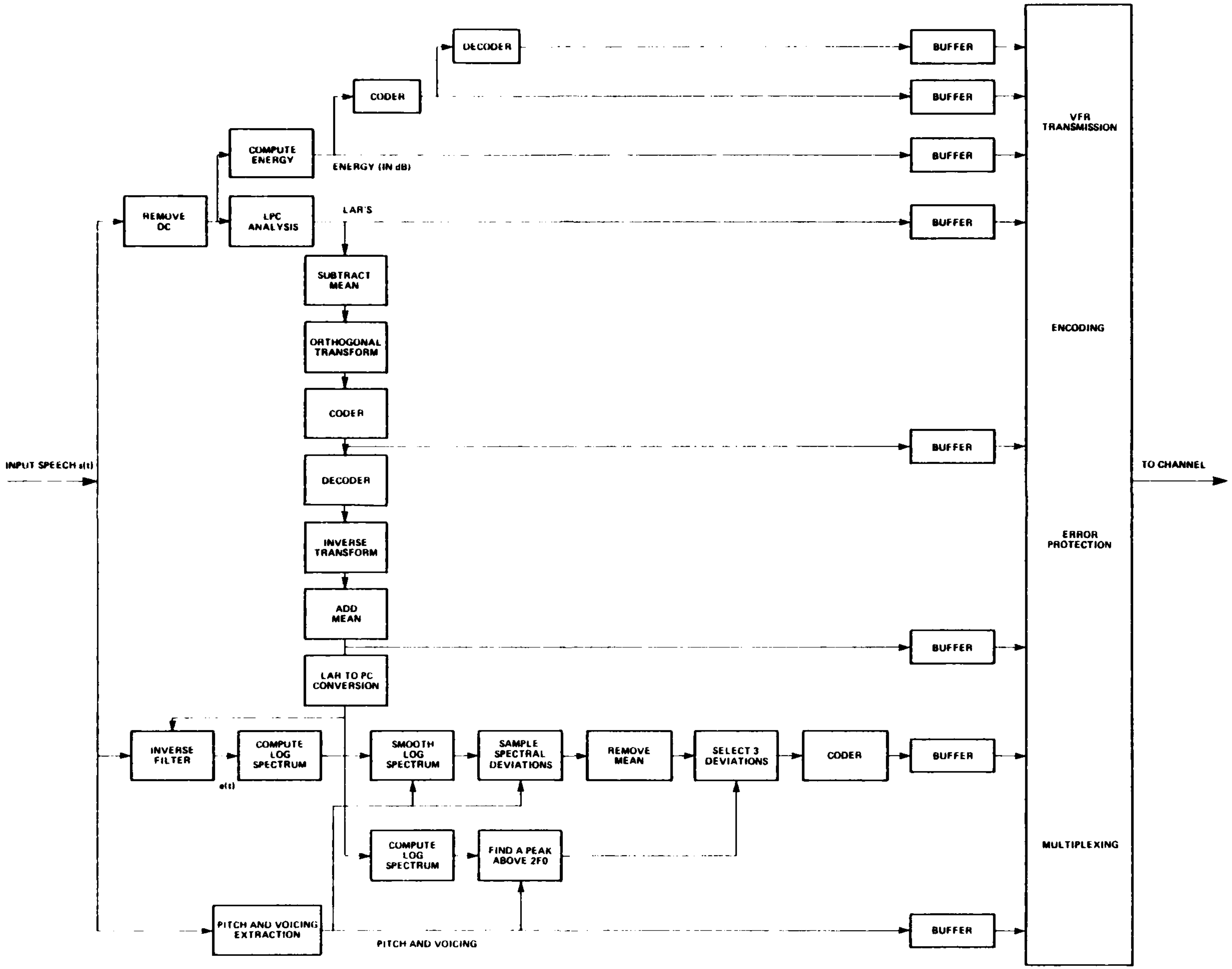


Fig. 10. The transmitter of the optimized 2.4 kb/s HDV coder.

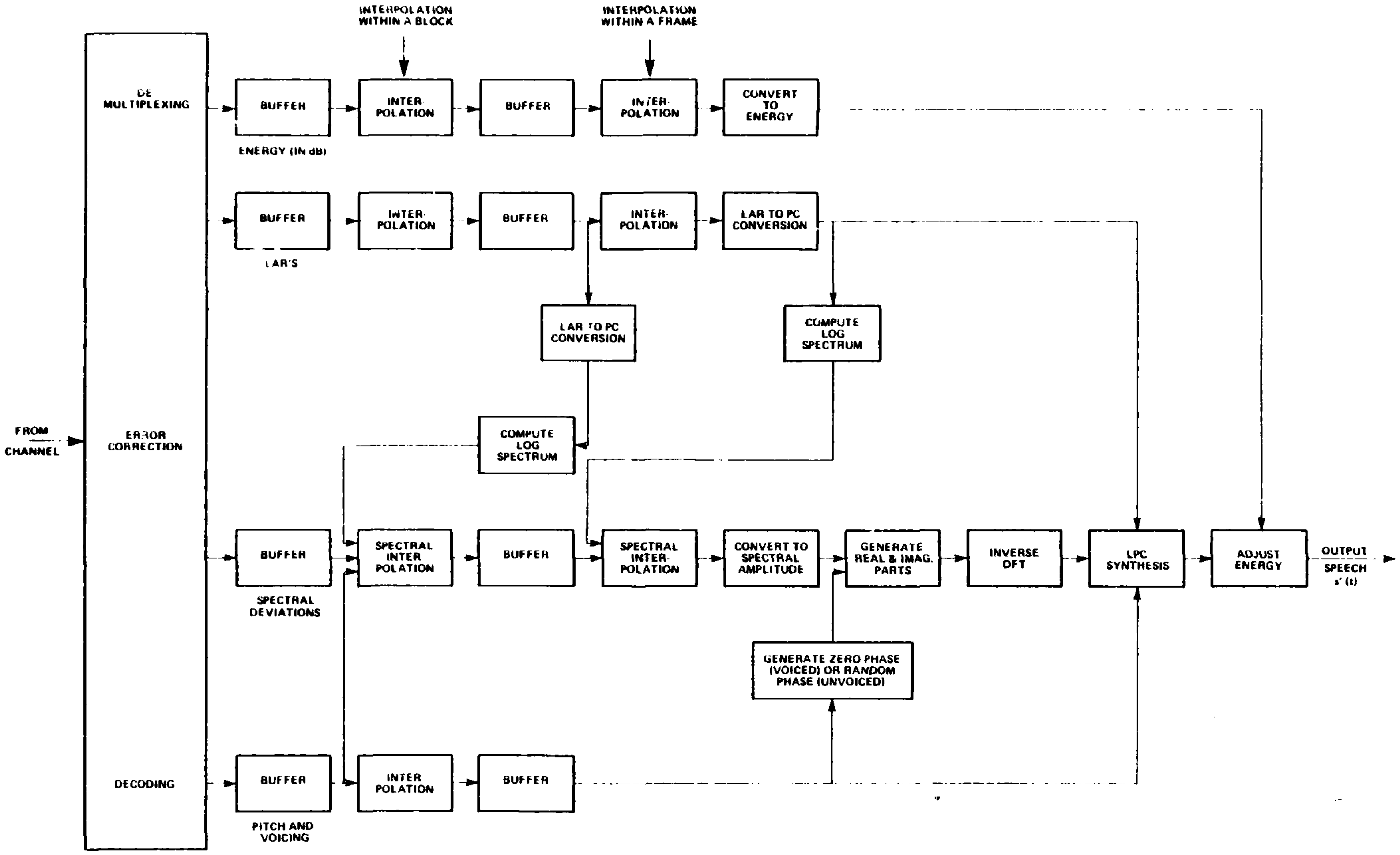


Fig. 11. The receiver of the optimized 2.4 kb/s HDV coder.

and coding the transformed coefficients using nonuniform coding tables. The coded LARs are decoded by decoding the transformed coefficients using nonuniform decoding tables, inverse transforming with the transpose of the eigenvector matrix, and adding the mean values. The extracted, coded, and decoded LARs are stored in three separate buffers, which are used in VFR transmission. The decoded LARs are converted to the decoded predictor coefficients (denoted as PC in Fig. 10, which are used in the spectral deviations analysis (see below). Pitch and voicing are extracted from the input speech using the modified AMDF-DYPTRACK algorithm (see Section 3.2 for a description of the modifications). The extracted pitch, which is already quantized to sixty levels, is stored along with the voicing bit in a buffer. The spectral deviations analysis consists of the following steps: obtain the residual e(t) by inverse filtering the input speech, with the inverse filter coefficients being the decoded predictor coefficients; compute the log spectrum of the residual by appending zeros and using a 1024-point FFT algorithm; smooth the log spectrum by computing the cepstrum (i.e., computing its DFT), windowing the cepstrum with a pitch-dependent window function (see Section 4.3), and inverse transforming the windowed ceptrum; compute spectral deviations by sampling the smoothed log spectrum of the residual at multiples of the pitch frequency F_0 if the analysis frame is voiced or at multiples of

100 Hz, if the frame is unvoiced; compute and remove the mean value from the deviations; compute the log spectrum of the LPC all-pole filter from the decoded predictor coefficients using 1024-point FFT; locate a peak in the log LPC spectrum above $2F_0$ (or above 200 Hz for unvoiced speech); and select 3 deviations for transmission as the first two and the one located just below the frequency of the peak located in the log LPC spectrum. The extracted 3 deviations are coded and stored in a buffer. After the analysis of 9 frames of input speech is completed, the block-encoded VFR scheme (see Section 9.1) is used to select six frames of parameter data for transmission. For each of six selected frames, the quantized parameter data, energy, LARs, pitch, voicing, and deviations, are binary encoded and error-protected (1) by using 3 Hamming (7,4) codewords to protect the 4 MSB's of energy, the 4 MSB's of pitch, and the 2 MSB's of each of the first two LARs, and (2) by using a 5-bit code for voicing, which is obtained by repeating the voicing bit 4 times. A synchronization bit is added to each frame. The block of 6 frames of data is multiplexed with three copies of a six-bit header code that identifies the selected six-frame sequence and transmitted over the channel.

At the receiver, shown in Fig. 11, the received data are demultiplexed, error-corrected, and decoded. The decoded parameters, energy (in dB), LARs, spectral deviations, pitch, and voicing, are stored in separate buffers and interpolated to generate the data of the untransmitted frames. All but spectral deviations are interpolated linearly, and spectral deviations are interpolated using the spectral interpolation method described in Section 9.5. The buffer that follows the interpolation in each of the four branches in Fig. 11 contains fixed-frame-rate data. For every frame, a second interpolation is performed, corresponding to the middle of the frame (10 ms), between its data and the data of the following frame. This within-frame interpolation is performed for energy, LARs, and spectral deviations. The next step is to convert energy in dB to linear scale, LARs to predictor coefficients, and deviations in dB to spectral amplitude. The spectral amplitudes corresponding to the deviations are combined with a zero phase for voiced speech or a random phase for unvoiced speech, to generate the real and imaginary parts of a DFT. An Inverse DFT produces a pitch-period (or 10 ms, for unvoiced speech) long excitation signal, which is then applied to the all-pole LPC synthesizer; the parameters of the synthesizer are set at the frame values for pitch periods within the first half of the frame and at the interpolated values for pitch periods in the second half. The energy of the synthesized speech is adjusted, every pitch period (or every 10 ms, for unvoiced speech), to be equal to the corresponding decoded or interpolated energy. The procedure used for this adjustment requires computing the output speech as the sum of the initial-condition response of the all-pole filter and an adjustable gain factor times its forced response. The digital output speech is passed through a D/A converter and an analog lowpass filter with its cutoff at 5 kHz to produce the analog output speech.

The optimized, 2.4 kb/s HDV coder produces noticeable speech-quality improvement over the 2.4 kb/s LPC coder. The improvement is in the form of reduced buzziness and background noises and a more natural voice quality. The extent of speech-quality improvement is more for male speakers than for female speakers. When operating over channels that cause 1% random bit-errors, the HDV coder produces some audible but mostly low-level distortions such as pops and clicks in the output speech; there is only a small difference in speech quaity and intelligibility between 1% channel error and noise-free cases.

15. FORTRAN SIMULATION OF THE OPTIMIZED CODER ON A PDP-11 MINICOMPUTER

During the project, we developed a general FORTRAN software package on our VAX-11/VMS computer, to simulate a number of different versions of the HDV coder. We extracted from this package the simulation of only the final optimized HDV coder, simplified it, and implemented it, using overlay programming techniques, on the sponsor's PDP-11/34 running under the RSX-11M operating system. The available address space on the PDP-11 is 32K 16-bit words. Of this space, approximately 12K words are used by the FORTRAN run-time library, leaving only about 2ØK words for user program and data.

15.1 Algorithm Simplifications

15.1.1 FFT Size

To reduce the storage (and computation), we reduced the size of all FFT operations required for spectral computations in the transmitter and the receiver from 1Ø24 to 256 points. We observed no perceivable change in the speech quality of the HDV coder as a result of this change.

15.1.2 Quantization of Spectral Deviations

We recall from Chapter 7 that the spectral deviations are quantized using 4-level uniform quantizers. The range of each quantizer is given by the 5- and 95-percentile points of the measured probability density functions of the corresponding spectral deviation. For the optimized HDV coder, the first two deviations are therefore quantized using fixed ranges, but the third transmitted deviation is quantized using a variable range since it can be located anywhere in the frequency range. To reduce the storage, we fixed the quantizer range for the third deviation at a value obtained by averaging over the ranges of spectral deviations at 3rd and higher harmonics. The range we chose is -7.5 dB to +5.5 dB. We found that fixing the quantizer range did not produce any speech-quality distortion.

15.2 FORTRAN Simulation on the VAX/VMS Computer

The general simulation software package that we developed on our VAX-11/VMS contained an interactive command structure in the main program and numerous options to simulate a number of different HDV coders. First, we deleted all the options and the associated mainline code and subroutines that are not part of the final system. Second, we removed the unnecessary buffers and reduced the sizes of a number of the required buffers. Third, we deleted all the interactive command structure and the associated code from the main program. We specified a number of coder parameters, flags, and switches via DATA statements. We verified the correctness of this simplified implementation by comparing its output against the output produced by our original development software.

15.3 Overlay Programming

We restructured the program to facilitate the use of the RSX-11M "Memory Resident Overlay" facility. This facility permits the entire user program to reside in main memory, assuming that the program is smaller than the physical size of memory. (On the sponsor's PDP-11, 124K words of main memory exist, of which 111K words are available to user programs.) Overlay segments are mapped into the user's 32K address space when they are referenced. (A subroutine is the smallest program unit that can comprise an overlay segment.) This method allows overlay segments to retain variable data from invocation to invocation.

Specifically, we decomposed the mainline program into subroutines to perform system initialization, analysis, variable frame rate block coding, channel simulation, parameter block decoding, and synthesis. (Each of these subroutines, of course, calls lower level routines for most of the computation.) We constructed a detailed overlay tree structure, and were able to reduce further the memory requirements of the largest branches of the tree. We continued and completed the task of "pruning" the tree so as to allow the final overlaid system to run in the allotted 32K address space.

In addition, we wrote routines to interface between our programs and the waveform I/O subroutines available on the sponsor's PDP-11. The final overlaid system was demonstrated on the sponsor's PDP-11.

REFERENCES

1. A.W.F. Huggins, R. Viswanathan and J. Makhoul, "Quality Ratings of LPC Vocoders: Effects of Number of Poles, Quantization, and Frame Rate," *IEEE International Conf. Acoustics, Speech and Signal Processing*, Hartford, CT, 1977, pp. 413-416.

2. R. Viswanathan, J. Makhoul and A.W.F. Huggins, "Speech Compression and Evaluation," Final Report, Contract No. MDA903-75-C-0180, Bolt Beranek and Newman Inc., BBN Report No. 3794, ADA055019, April 1978.

3. J. Makhoul, R. Viswanathan, R. Schwartz and A.W.F. Huggins, "A Mixed-Source Model for Speech Compression and Synthesis," *J. Acoust. Soc. Amer.*, Vol. 64, Dec. 1978, pp. 1577-1581.

4. B.S. Atal and N. David, "On Synthesizing Natural-Sounding Speech by Linear Prediction," *IEEE International Conf. Acoustics, Speech and Signal Processing*, Washington, DC, April 1979, pp. 44-47.

5. J.Makhoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE*, Vol. 63, April 1975, pp. 561-580.

6. J. Makhoul and J. Wolf, "Linear Prediction and the Spectral Analysis of Speech," Report No. 2304, Bolt Beranek and Newman Inc., Cambridge, MA, Aug. 1972.

7. Xavier Rodet, IRCAM, Paris, France, Personal communication, April 1979.

8. R.D. Leites and V.N. Sobolev, "The Influence of Phase Effect on the Aural Perception of Synthesized Speech," Telecommunications and Radio Engineering, 28/29 No. 1, pp. 50-53, 1974.

9. R.B. Monsen and A.M. Engebretson, "Study of Variations in the Male and Female Glottal Wave," *J. Acoust. Soc. Amer.*, Vol. 62, Oct. 1977, pp. 981-993.

10. J. Makhoul, "Spectral Linear Prediction: Properties and Applications," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-23, No. 3, June 1975, pp. 283-296.

11. B.S. Atal and S.L. Hanauer, "Speech Analysis and Synthesis by Linear Prediction of the Speech Wave," J. Acoust. Soc. Amer., Vol. 50 1971, pp. 637-655.

12. T.E. Tremain, J.W. Fussell, R.A. Dean, B.M. Abzug, M.D. Cowing and P.W. Boudra, Jr., "Implementation of Two Real-Time Narrowband Speech Algorithms," Proc. EASCON '78, Washington, D.C., September 1978, pp. 698-708.

13. R.W. Schafer and L.R. Rabiner, "System for Automatic Formant Analysis of Voiced Speech," J. Acoust. Soc. Amer., Vol. 47, No. 2, Feb. 1970, pp. 634-648.

14. N.J. Miller, "Pitch Detection by Data Reduction," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-23, Feb. 1975, pp. 72-79.

15. R. Zelinski and P. Noll, "Adaptive Transform Coding of Speech Signals," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-25, August 1977, pp. 299-309.

16. M. Berouti and J. Makhoul, "An Embedded-Code Multirate Speech Transform Coder," Proc. 1980 Int. Conf. Acoustics, Speech, and Signal Processing, Denver, CO, April 1980, pp. 356-359.

17. R. Viswanathan and J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-23, June 1975, pp. 309-321.

18. J. Max, "Quantizing for Minimum Distortion," IRE Trans. Info. Theory, Vol. IT-6, March 1960, pp. 7-12.

19. A. Segall, "Bit Allocation and Encoding for Vector Sources," IEEE Trans. Inform. Theory, Vol. IT-22, March 1976, pp. 162-169.

20. M.R. Sambur, "An Efficient Linear Prediction Vocoder," Bell Syst. Tech. J., Vol. 54, pp. 1693-1723, Dec. 1975.

21. J.W. Fussell, "The Karhunen-Loeve Transform Applied to the Log Area Ratios of a Linear Predictive Speech Coder," Int. Conf. Acoustics, Speech, and Signal Processing, Denver, CO, April 1980, pp. 36-39.

22. A. Buzo, A.H. Gray Jr., R.M. Gray, and J.D. Markel, "Speech Coding Based Upon Vector Quantization," IEEE Trans. Acoustics, Speech, and Signal Processing, Vol. ASSP-28, Oct. 1980, pp. 562-574.

23. J. Makhoul, S. Roucos, M. Krasner, R. Schwartz, and J. Sorensen, "Research on Narrowband Communications," Quarterly Progress Report, BBN Report No. 4620, March 1981.

24. J.W. Fussell, "A 1200 Bit Per Second Voice Coder Based on Differential Linea Prediction," Doctoral Thesis, The School of Eng. and Applied Science, The George Washington University, Feb. 1981.

25. R. Viswanathan, J. Makhoul and R. Wicke, "The Application of a Functional Perceptual Model of Speech to Variable-Rate LPC Systems," IEEE International Conf. Acoustics, Speech and Signal Processing, Hartford, CT, May 1977, pp. 219-222.

26. E. Blackman, R. Viswanathan, W. Russell and J. Makhoul, "Narrowband LPC Speech Transmission over Noisy Channels," IEEE International Conf. Acoustics, Speech and Signal Processing, Washington, D.C., April 1979, pp. 60-63.

27. R. Viswanathan, W. Russell, E. Blackman, A. Higgins, and J. Makhoul, "YOHO Variable Rate LPC Study," Final Report, BBN Report No. 4568, Jan. 1981 (Limited distribution).

28. R. Viswanathan, J. Makhoul, R. Schwartz, and A.W.F. Huggins, "Variable-Frame-Rate Transmission: A Review of the Methodology and Application to Narrowband LPC Speech Coding," Accepted for publication in IEEE Trans. Comm., April 1982.

29. R. Viswanathan, E. Blackman, J. Makhoul, and W. Russell, "YOHO Variable Rate LPC Study," Final Report, BBN Report No. 3617, Aug. 1977 (Limited distribution).

30. J.W. Fussell, T.E. Tremain, P.W. Boudra, Jr., and M.D. Cowing, "Providing Channel Error Protection for a 2400 BPS Linear Predictive Coded Voice System," IEEE International Conf. Acoustics, Speech and Signal Processing, April 1978, pp. 462-465.

DATE

ILME